

**PORTABLE ELECTROCHEMICAL IMPEDANCE SPECTROSCOPY
MEASUREMENT DEVICE FOR BATTERY QUALITY CHECK.**

By
Chan Jian Hong

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)
COMPUTER ENGINEERING
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2025

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts Dr. Lee Wai Kong, for his guidance, patience, and support throughout the development of my project, "Portable Electrochemical Impedance Spectroscopy Measurement Device for Battery Quality Check." His insights have been invaluable in shaping this project, and I am truly grateful for his mentorship.

A very special thanks to my parents and family for supporting and motivating me throughout my academic journey. Their patience, sacrifices and love is the driving force for me to keep moving and achieve success in life.

COPYRIGHT STATEMENT

© 2025 Chan Jian Hong. All rights reserved.

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Technology (Honours) Computer Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ABSTRACT

This project presents a portable Electrochemical Impedance Spectroscopy (EIS) device for real-time lithium-ion battery (LIB) health assessment, addressing the accuracy, cost, and deployability limits of conventional methods such as Coulomb counting and open-circuit voltage analysis. The prototype integrates a miniaturized impedance front end with an embedded controller, delivering 19-point frequency sweeps and 1 kHz sampling while rendering live Nyquist plots on a touch display and logging results to microSD in CSV format. The system emphasizes practical field use: auto-scaled visualization, non-blocking timing for looped measurements, and a queue-based logger that maintains responsiveness under sustained writes. Repeatability tests show stable overlays for the same cell and clear separation between different cells, indicating reliable acquisition for downstream analytics. In parallel, the design establishes an AI-ready pipeline by structuring data for feature extraction and model development to refine State of Charge (SoC) and State of Health (SoH) estimation. Preliminary analysis supports the feasibility of on-device or near-edge inference using lightweight models to enable predictive maintenance, early degradation detection, and optimized charging strategies. By combining portability, real-time operation, and data-driven diagnostics, the proposed EIS platform offers a cost-effective path toward proactive battery management in electric vehicles, renewable energy storage, and consumer electronics.

Area of Study: Embedded System, Battery Management Systems

Keywords: Electrochemical Impedance Spectroscopy, Lithium-Ion Battery Diagnostics, AI-Driven Predictive Maintenance, Embedded System Design, Arduino IDE, Real-Time Processing, State of Charge Estimation, State of Health Monitoring, Edge Computing, On-Device Analytics.

TABLE OF CONTENTS

TITLE PAGE	i
ACKNOWLEDGEMENTS	ii
COPYRIGHT STATEMENT	iii
ABSTRACT	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xii
LIST OF SYMBOLS	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1 INTRODUCTION	16
1.1 Problem Statement and Motivation	16
1.2 Project Scope	17
1.3 Project Objective	17
1.4 Contributions	18
1.5 Background of Study	19
1.5 Report Organization	23
CHAPTER 2 LITERATURE REVIEW	25
2.1 Previous Works on Electrochemical Impedance Spectroscopy for Battery Diagnostic	25
2.1.1 Existing Portable EIS Solutions	25
2.1.2 AI Integration in EIS Analysis	26
2.1.3 Predictive Maintenance Using EIS	28
2.2 Limitations of the Previous Studies	29
2.3 Proposed Solutions	30
CHAPTER 3 PROPOSED METHOD/APPROACH	33
3.1 System Design Diagram	33
3.1.1 System Architecture Diagram	33
3.1.2 User Case Diagram and Description	34
3.1.3 Activity Diagram	36
CHAPTER 4 SYSTEM DESIGN	37

4.1	System Block Diagram	37
4.2	System Components Specifications	38
4.2.1	Hardware	38
4.2.2	Software	44
4.3	Circuits and Components Design	45
4.4	System Components Interaction Operation	46
CHAPTER 5 SYSTEM IMPLEMENTATION		48
5.1	Hardware Setup	48
5.2	Software Setup	51
5.2.1	Arduino Libraries Used	51
5.2.2	Source File Organization	52
5.2.3	Ohm Symbol integration	53
5.2.4	LVGL Image Asset Integration	53
5.2.5	SD Initialization	54
5.3	Setting and Configuration	55
5.4	System Operation	58
5.5	Implementation Issues and Challenges	77
5.6	Concluding Remark	78
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION		79
6.1	System Testing and Performance Metrics	79
6.2	Testing Setup and Result	80
6.2.1	Timing correctness in Modes 3 and 4	80
6.2.2	Logging integrity shown by CSV evidence	81
6.2.3	Throughput and table-to-CSV cross-check in Modes 5 and 6	82
6.2.4	Repeatability	83
6.3	Project Challenges	83
6.4	Objectives Evaluation	84
6.5	Concluding Remark	84
CHAPTER 7 CONCLUSION AND RECOMMENDATIONS		85
7.1	Conclusion	85
7.2	Recommendation	87
REFERENCES		89
APPENDIX		92

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	Nyquist plots for battery states at 30%, 70% and 90%	19
Figure 1.1	Battery impedance measurement and modelling for diagnosing and preventing electric vehicle battery issues.	21
Figure 3.1	System Architecture Diagram for the System	33
Figure 3.2	User Case Diagram for the System	34
Figure 3.3	Activity Diagram of the System	36
Figure 4.1	System Block Diagram	37
Figure 4.2	ESP32-S3 Interface Board Between DNB110xA and JC4827W543	39
Figure 4.3	DNB110xA	40
Figure 4.4	4.3-inch LCD Display	41
Figure 4.5	18650 Lithium Battery Cell	42
Figure 4.6	SD Card Module for Data Logging	43
Figure 4.7	Arduino Logo	44
Figure 4.8	LVGL Logo	44
Figure 4.9	Circuit and pin-level interconnect of the portable EIS device.	45
Figure 4.10	Flowchart of the System	46
Figure 5.1	Hardware Setup. Battery fixture, ESP32-S3 interface board with DNB110xA, and JC4827W543 display showing the LVGL “EIS Battery Monitor” screen	49
Figure 5.2	Libraries Used	51
Figure 5.3	Project Source Code	52

Figure 5.4	LVGL snippet rendering the Ω symbol	53
Figure 5.5	LVGL snippet displaying the Malaysia Flag from a converted C-array image	53
Figure 5.6	SD initialization on boot showing successful mount at 10 MHz	54
Figure 5.7	Arduino IDE settings for ESP32-S3	55
Figure 5.8	EIS Battery Monitor v9.3 UI	56
Figure 5.9	SD card File Structure	57
Figure 5.10	Mode not selected warning	58
Figure 5.11	No device detected warning	59
Figure 5.12	Mode 1 selection screen	60
Figure 5.13	Sweep progress shown	60
Figure 5.14	Sweep complete status	61
Figure 5.15	Mode 2 selection with Avg X	62
Figure 5.16	Averaging sweep 1 of X	62
Figure 5.17	Averaging sweep 2 of X	63
Figure 5.18	Averaging sweep 3 of X	63
Figure 5.19	Averaging sweep X of X complete	64
Figure 5.20	Averaged Nyquist result	64
Figure 5.21	Mode 3 selection screen	65
Figure 5.22	Continuous mode started	65
Figure 5.23	Waiting between cycles	66
Figure 5.24	Plot after recent cycle	66
Figure 5.25	Mode 4 selection with Avg X	67

Figure 5.26	Average loop sweep 1 of X	68
Figure 5.27	Figure 5.27: Average loop sweep 2 of X	68
Figure 5.28	Figure 5.28: Average loop sweeps X of X	69
Figure 5.29	Figure 5.29: Countdown to next cycle	69
Figure 5.30	Averaged loop result	70
Figure 5.31	Mode 5 selection screen	71
Figure 5.32	Live stream started	71
Figure 5.33	FStreaming samples in real time	72
Figure 5.34	Live sample table view	72
Figure 5.35	Mode 6 selection with Avg X	73
Figure 5.36	1 kHz Average Dialog Initializing	74
Figure 5.37	Sample 1 of X in progress	74
Figure 5.38	Sample X of X complete	75
Figure 5.39:	Averaged point plot in chart	75
Figure 5.40	Averaging table page 1	76
Figure 5.41	Averaging table page 2	76
Figure 6.1	Serial Monitor countdown confirming the one-minute dwell timer.	80
Figure 6.2	On-screen dialog showing the same waiting seconds for Mode 3/4.	81
Figure 6.3	CSV excerpt: third-frequency record showing Real/Imag values used for cross-check.	81
Figure 6.4	Device display: the third-frequency point reports the same Real/Imag values as Figure 6.3	82

Figure 6.5	Table on device matches the corresponding CSV rows for the listed 1 kHz samples.	82
Figure 6.6	Nyquist plots for two cells	83

LIST OF TABLES

Table Number	Title	Page
Table 4.1	Specification of Interface Board	39
Table 4.2	Specification of DNB110xA	40
Table 4.3	Specification of JC4827W543 HMI Display with Integrated ESP32-S3-WROOM-1	41
Table 4.4	Specification of 18650 Lithium Battery	42
Table 4.5	Specification of LCD Display	43
Table 5.1	Pin Connection of ESP32-S3 and DNB110Xa	50
Table 5.2	Pin Connection of ESP32-S3 and GT911 Touch Driver	50
Table 5.3	Pin Connection of ESP32-S3 and microSD (HSPI)	50

LIST OF SYMBOLS

Hz	Hertz (frequency)
Ω	Ohm (resistance)

LIST OF ABBREVIATIONS

<i>ANNs</i>	Artificial Neural Network
<i>BMS</i>	Battery Management System
<i>CRC</i>	Cyclic Redundancy Check
<i>CSV</i>	Comma-Separated Values
<i>ECMs</i>	Equivalent Circuit Models
<i>EIS</i>	Electrochemistry Impedance Spectroscopy
<i>EV</i>	Electric Vehicle
<i>FPGAs</i>	Field-Programmable Gate Arrays
<i>GPR</i>	Gaussian Process Regression
<i>GUI</i>	Graphic User Interface
<i>I²C</i>	Inter-Integrated Circuit
<i>LAM</i>	Loss of Active Materials
<i>LIBs</i>	Lithium-ion Batteries
<i>LSTM</i>	Long Short-Term Memory
<i>LVGL</i>	Light and Versatile Graphics Library
<i>ML</i>	Machine Learning
<i>OCV</i>	Open Circuit Voltage
<i>PSRAM</i>	Pseudo-Static RAM
<i>RTOS</i>	Real-Time Operating System
<i>RUL</i>	Remaining Useful Life
<i>SD</i>	Secure Digital (memory card)
<i>SEI</i>	Solid Electrolyte Interphase
<i>SOC</i>	State-of-Charge
<i>SOH</i>	State-of-Health
<i>TPUs</i>	Tensor Processing Units
<i>UART</i>	Universal Asynchronous Receiver-Transmitter
	Usage Faults

User Interface

CHAPTER 1

Introduction

In this chapter, we present the background, problem statements and motivations of our project and our contributions to the field of battery diagnostics and predictive maintenance.

1.1 Problem Statement and Motivation

Conventional methods for monitoring Lithium-Ion Batteries (LIBs) such as Coulomb counting, open-circuit voltage (OCV) measurements, and equivalent circuit models lack accuracy and real-time capability. These techniques suffer from issues like sensor drift, inaccuracies, and an inability to capture the full complexity of battery behavior under dynamic operating conditions. As a result, precise estimation of the SoC and SoH, which are critical for optimizing battery performance and longevity, is hindered. This limitation poses significant challenges in ensuring battery safety and preventing failures in critical applications like electric vehicles (EVs) and energy storage systems.

In addition, while Electrochemical Impedance Spectroscopy (EIS) has proven to be an effective tool for battery diagnostics, existing systems are bulky, expensive, and impractical for real-time monitoring in real-world applications. These laboratory-grade systems require controlled environments and sophisticated data analysis, making them unsuitable for on-site diagnostics in applications such as EVs and distributed energy storage. Although portable EIS solutions exist, they often lack the necessary precision and computational power to accurately assess battery health in practical settings, limiting their widespread use.

The motivation behind this project stems from the need to address these limitations and develop a portable, AI-enhanced EIS system that provides real-time, accurate battery diagnostics. Traditional methods for SoC and SoH estimation are insufficient, and there is a growing demand for a more advanced solution that improves the reliability, performance, and lifespan of LIBs. Integrating AI and machine learning with EIS technology can enhance the accuracy of battery health monitoring, enabling predictive maintenance and reducing unexpected failures. This project aims to bridge the gap

between existing, bulky EIS systems and the need for a compact, cost-effective solution that can be applied in real-world, on-site conditions, ultimately contributing to the sustainability and safety of modern energy storage technologies.

1.2 Project Scope

This project aims to develop a Portable Electrochemical Impedance Spectroscopy (EIS) Measurement Device designed to assess the SoH of LiBs. The device will integrate machine learning algorithms to analyze the battery's internal resistance, voltage, and temperature, providing a more accurate prediction of battery performance and extending its lifespan. By enabling real-time monitoring, predictive diagnostics, and preventative maintenance, this system will address critical challenges in ensuring battery reliability and performance in applications such as electric vehicles (EVs), renewable energy storage, and consumer electronics.

The scope of the project involves creating a compact, cost-effective, and portable EIS system with an interactive user interface (UI). This UI will display real-time impedance data, allowing users to monitor and interact with battery health assessments. Machine learning models will be applied to improve the accuracy of SoC and SoH predictions. The goal is to provide a scalable solution that offers not only immediate feedback on battery conditions but also predictive insights to optimize performance and prevent failure. The final deliverable will be a fully functional portable EIS system, complete with a user-friendly interface and integrated AI capabilities for advanced battery diagnostics.

1.3 Project Objective

The primary objective of this project is to design and develop a compact and portable Electrochemical Impedance Spectroscopy (EIS) measurement system that enables high-accuracy, real-time battery diagnostics.

The main objective can be divided into a few sub-objectives as follows:

- To design and develop a compact and portable EIS measurement system that provides high-accuracy battery diagnostics in real-time.

- To develop an interactive user interface (UI) that allows users to monitor impedance measurements, voltage, temperature, and other key battery health indicators.
- To extract key EIS features and log data to microSD (CSV) for offline analysis and future ML model refinement.

1.4 Contributions

This project aims to make significant contributions to the field of battery diagnostics and predictive maintenance by developing a portable and AI-enhanced EIS measurement system. Unlike traditional laboratory-grade EIS systems, which are often bulky, expensive, and impractical for on-site applications, this project introduces a cost-effective, compact, and real-time diagnostic tool for LIB health monitoring. By integrating ML algorithms with EIS data analysis, the system will enhance the accuracy of SoC and SoH estimation, offering a more reliable alternative to conventional diagnostic methods. This shift will enable industries such as electric vehicles, renewable energy storage, and consumer electronics to adopt a more proactive approach to battery management, ultimately reducing maintenance costs, preventing premature battery failures, and extending battery lifespan.

The significance of this project lies in its potential to enhance battery management strategies, making them more proactive and data-driven. This project addresses a critical gap in the current market by providing a portable and scalable solution that improves the accuracy of battery health assessments. The integration of machine learning with Electrochemical Impedance Spectroscopy (EIS) allows for real-time analysis and more reliable predictions, offering a practical, accessible alternative to existing diagnostic methods. As the world increasingly shifts towards renewable energy solutions and electric vehicles, this innovation becomes increasingly important, ensuring that energy storage systems are more efficient, reliable, and sustainable. By offering a portable, scalable, and precise solution for battery diagnostics, this project has the potential to transform how industries manage and maintain battery-powered systems, thus creating long-term benefits for both users and society.

1.5 Background of Study

Electrochemical Impedance Spectroscopy (EIS)

Electrochemical Impedance Spectroscopy (EIS) is a non-destructive analytical technique widely used for studying the electrochemical behaviour of materials and systems, particularly LIBs [6]. It provides critical insights into battery performance, aging mechanisms, and electrochemical properties by measuring the impedance response of a system over a range of frequencies. Unlike conventional diagnostic techniques that rely solely on voltage, current, and temperature measurements, EIS offers a more comprehensive approach by capturing dynamic impedance variations that correspond to electrochemical reactions within the battery. This technique has become an indispensable tool for battery research and development, enabling scientists and engineers to understand charge transfer kinetics, detect internal resistance variations, and optimize battery material compositions [1].

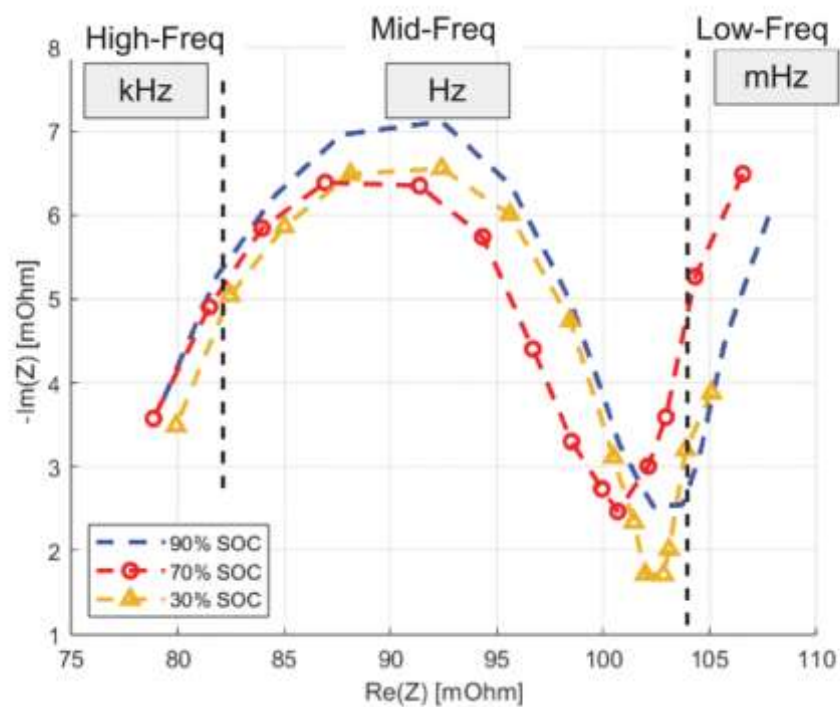


Figure 1.1: Nyquist plots for battery states at 30%, 70% and 90% is adapted from [1].

The application of EIS in lithium-ion battery analysis involves applying a small AC voltage or current perturbation across a battery and analysing the resulting response in terms of impedance. This impedance is then transformed into the frequency domain using mathematical techniques such as the Fourier transform, which decomposes the

signal into its constituent frequency components [2]. By studying the real and imaginary parts of impedance, researchers can construct Nyquist plots that reveal critical information about electrolyte conductivity, charge transfer resistance, and solid electrolyte interphase (SEI) layer behaviour. This ability to track changes in internal resistance and reaction kinetics makes EIS particularly useful for diagnosing battery degradation mechanisms and optimizing charging and discharging protocols for extended cycle life and reliability. Recent advancements in EIS technology have enabled real-time impedance measurements in operational battery systems, making it a powerful tool for both laboratory and industrial applications [3].

State of Charge (SoC) and State of Health (SoH)

SoC and SoH are fundamental parameters for assessing battery performance and longevity. SoC represents the available charge within a battery as a percentage of its total capacity, serving as an indicator of how much energy remains before recharging is required. Meanwhile, SoH measures the overall condition of a battery relative to its initial state, providing valuable insights into degradation trends and remaining useful life. Accurate estimation of these parameters is essential for optimizing battery management systems (BMS), enhancing safety, and ensuring efficient utilization of energy storage systems. Traditional SoC and SoH estimation methods, such as Coulomb counting and open-circuit voltage (OCV) analysis, have inherent limitations due to sensor inaccuracies, temperature dependencies, and cumulative errors over time [1], [4]. EIS has emerged as a highly effective technique for improving the accuracy of SoC and SoH estimation. Unlike conventional methods, EIS captures electrochemical signatures that directly reflect changes in battery impedance, providing a more reliable assessment of a battery's internal state. Studies have demonstrated that machine learning (ML) models trained on EIS data can significantly enhance SoC and SoH prediction accuracy. For example, deep learning techniques, including artificial neural networks (ANNs) and long short-term memory (LSTM) networks, have been applied to analyze EIS-derived impedance spectra, achieving prediction accuracies exceeding 93% [3]. Furthermore, equivalent circuit models (ECMs) have been used to convert EIS data into low-dimensional representations, enabling efficient real-time SoC and SoH classification

with minimal computational overhead. These advancements highlight the potential of integrating EIS with AI-driven analytics to develop next-generation battery monitoring systems capable of adaptive learning and predictive diagnostics.

Predictive Maintenance and EIS Applications

Predictive maintenance has emerged as a transformative approach for ensuring the long-term reliability and efficiency of battery-powered systems. Traditional maintenance practices often rely on reactive or scheduled maintenance routines, which can be inefficient and costly. In contrast, predictive maintenance leverages real-time monitoring techniques such as EIS to assess battery health continuously and predict potential failures before they occur. By analysing impedance data trends over time, predictive maintenance systems can identify early signs of degradation, allowing for timely intervention and optimization of battery operation. This proactive approach is particularly beneficial in applications where battery reliability is critical, such as electric vehicles, aerospace, and grid-scale energy storage systems [5].

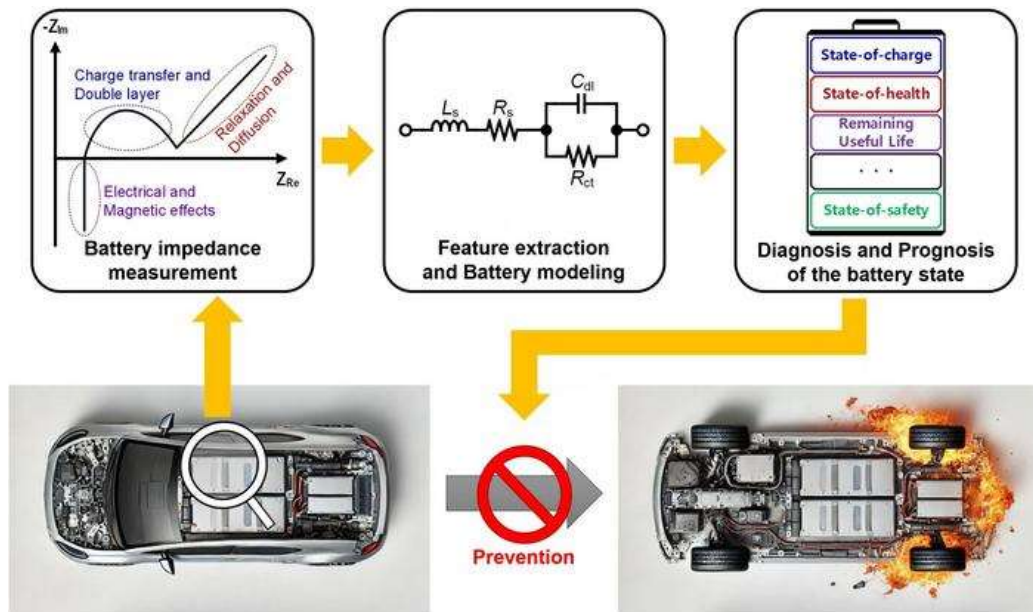


Figure 1.1: Battery impedance measurement and modelling for diagnosing and preventing electric vehicle battery issues is adapted from [15].

From [15], EIS evaluates both the impedance magnitude and its fluctuations within a battery to assess the battery efficiency and loss. Hence, SoC and SoH are essential tools for evaluating battery efficiency. They can also be used to identify thermal

characteristics, detect chemical or physical changes, predict battery health, and identify the root cause of failures. The integration of EIS with advanced predictive maintenance frameworks has paved the way for intelligent battery management solutions that adapt to changing operational conditions. Machine learning algorithms trained on historical EIS data can detect patterns indicative of performance deterioration and provide actionable insights for battery optimization. For example, predictive models can recommend adjustments to charging protocols, temperature management strategies, and load balancing techniques to extend battery life and enhance overall system efficiency. Additionally, predictive maintenance systems equipped with remote monitoring capabilities enable fleet operators and energy providers to manage large-scale battery deployments more effectively, reducing operational costs and minimizing downtime. As EIS technology continues to evolve, its role in predictive maintenance is expected to expand, driving innovations in sustainable energy storage and smart battery management solutions [6].

Importance of Portable EIS Systems

EIS has been widely recognized as a crucial diagnostic tool for lithium-ion batteries, offering real-time insights into battery performance, degradation mechanisms, and state estimation. However, traditional EIS measurement systems are often large, expensive, and confined to laboratory environments, limiting their accessibility for on-site battery monitoring and real-world applications [1]. This restriction poses significant challenges for industries relying on batteries, such as electric vehicle (EV) manufacturers, renewable energy storage providers, and consumer electronics companies. Without portable EIS systems, battery health diagnostics must be conducted in controlled laboratory conditions, requiring extensive setup and costly instrumentation. As a result, real-time monitoring and field-based diagnostics remain impractical, preventing proactive maintenance strategies that could enhance battery longevity and prevent unexpected failures.

The development of portable EIS measurement systems has emerged as a game-changer, enabling on-site, real-time battery diagnostics without relying on bulky laboratory equipment. Recent advancements in miniaturized impedance analyzers, AI-driven data

processing, and cost-effective hardware components have made it feasible to integrate EIS technology into compact, handheld, and even embedded battery management systems (BMS) [2]. Portable EIS devices allow for rapid battery health assessments in electric vehicle fleets, smart grids, and industrial energy storage applications, reducing downtime and maintenance costs. Additionally, integrating EIS into battery-powered applications ensures that critical battery diagnostics can be performed without the need for extensive disassembly or transportation to specialized testing facilities. By making EIS portable, industries can implement real-time health monitoring and predictive maintenance strategies, ultimately leading to safer, more reliable, and longer-lasting battery systems.

1.5 Report Organization

This report is structured into several chapters to outline the development process of portable electrochemical impedance spectroscopy.

- **Chapter 1: Introduction**

This chapter provides an overview of the project, including the problem statement, motivation, and objectives. It introduces the concept of electrochemical impedance spectroscopy and its importance for battery diagnostics, particularly for portable applications in electric vehicles and energy storage systems.

- **Chapter 2: Literature Review**

This chapter reviews previous works and existing systems related to electrochemical impedance spectroscopy, focusing on the challenges and limitations of current technologies. It highlights advancements in portable EIS solutions, AI integration, and predictive maintenance using EIS, and identifies gaps in existing methods that the current project aims to address.

- **Chapter 3: Proposed Method**

In this chapter, the overall approach for the portable EIS system is outlined. It describes the measurement workflow, operating modes, frequency plan, and data flow from acquisition to plotting and logging. The methodology for

impedance data handling, feature extraction readiness, and real-time processing is also detailed.

- **Chapter 4: System Design**

This chapter presents the detailed hardware and software design. It covers the ESP32-S3 controller, DNB110xA front end, JC4827W543 touch display, power and pin mapping, and communication protocols. The firmware stack (LVGL UI, FreeRTOS tasking, UART/HSPI, SD logging) is described together with design challenges and how they were resolved.

- **Chapter 5: Implementation and System Operation**

This chapter documents the assembled setup, configuration steps, and run-time behavior across Modes 1–6. It explains the user interface flow, auto-scaling charts, error handling, and CSV file organization, providing a reproducible guide to operate the device.

- **Chapter 6: System Evaluation and Discussion**

This chapter presents the testing methodology, performance metrics, and key results using serial traces, LVGL screen observations, and CSV evidence. It discusses responsiveness, stability, logging integrity, and repeatability, and summarizes project challenges encountered during validation.

- **Chapter 7: Conclusion and Recommendations**

This chapter summarizes the achievements, limitations, and lessons learned. It provides recommendations for future work, including benchtop accuracy benchmarking, frequency-set refinement, lightweight on-device feature extraction, and data pipelines for ML-based SoC/SoH diagnostics.

CHAPTER 2

Literature Review

2.1 Previous Works on Electrochemical Impedance Spectroscopy for Battery Diagnostic

This section explores a range of previous studies and current technologies that underline the effectiveness in analysing SoC and SoH of LIBs using EIS system. Few advancements have been made to enhance battery diagnostics, with a strong focus on improving portability, AI integration, and predictive maintenance capabilities.

2.1.1 Existing Portable EIS Solutions

EIS has emerged as a powerful technique for analysing electrochemical properties across various applications, including battery diagnostics, fuel cell performance assessment, and biological sensing. Traditionally, EIS has been performed using benchtop instruments such as LCR meters and impedance analysers, which offer high precision but are often expensive, bulky, and unsuitable for field applications. As a result, the development of portable EIS systems has been an active research area aimed at enabling real-time, on-site electrochemical measurements without reliance on laboratory environments.

Recent projects in microcontroller-based portable EIS devices have demonstrated their capability to offer cost-effective and efficient solutions. For instance, a low-cost portable EIS system was introduced, utilizing a microcontroller-based board integrated with an external waveform generator for impedance measurement in liquid and semi-liquid media. This system effectively measured impedance spectra for saline solutions, with a high determination coefficient ($R^2 = 0.9965$), thereby confirming its accuracy and feasibility for real-world applications [7]. The use of 3D-printed structures further improved portability, making it suitable for in-field measurements.

Furthermore, the development of on-chip impedance analysers such as the AD5933 by Analog Devices has significantly contributed to the miniaturization of EIS devices. These analysers offer an impedance measurement range of 1 k Ω to 10 M Ω , with frequency ranges between 1 kHz and 100 kHz, making them ideal for portable

applications [8]. Researchers have successfully integrated these chips into wearable and smartphone-based impedance measurement systems, enabling real-time health monitoring and electrochemical sensing [9]. However, despite their affordability and portability, these devices often suffer from limitations in terms of accuracy and frequency range compared to traditional laboratory-based EIS analysers.

In the field of energy storage and power electronics, portable EIS systems have been explored for in situ monitoring of lithium-ion batteries and fuel cells. Electrochemical impedance spectroscopy has been employed to assess the SoC of lithium-ion batteries using portable measurement systems, with Gaussian Process Regression (GPR) models achieving an error margin below 3.8% [10]. Similarly, researchers have developed portable EIS solutions for fuel cell characterization, where impedance measurements of different cathode flow-field designs were utilized to optimize fuel cell performance and identify degradation mechanisms in open-cathode polymer electrolyte membrane fuel cell (OC-PEMFC) stacks [11].

While portable EIS solutions have significantly improved accessibility and ease of use, challenges remain in achieving the precision and versatility of laboratory-grade systems. Future advancements should focus on enhancing the frequency range, signal stability, and integration with AI-based analytical models for automated data interpretation.

2.1.2 AI Integration in EIS Analysis

AI has been increasingly integrated into EIS analysis to enhance data interpretation, improve diagnostic accuracy, and automate the identification of electrochemical phenomena. Traditional EIS data analysis involves fitting impedance spectra to equivalent circuit models, which requires expertise and extensive computational resources. AI-driven approaches, such as machine learning and deep learning models, offer the potential to automate this process while improving accuracy and efficiency.

One of the major areas of AI integration in EIS is in battery health diagnostics. By leveraging AI algorithms, researchers have developed models that can estimate battery degradation modes, such as loss of lithium inventory (LLI) and loss of active materials

(LAM), with high precision. A hybrid approach combining impedance-based modelling and AI was introduced for real-time degradation diagnosis in lithium-ion batteries, achieving an error margin of less than 0.5% for capacity fade estimation [12]. Similarly, AI-based fuzzy clustering algorithms have been employed for durability analysis of fuel cells, distinguishing between normal and abnormal operating conditions based on impedance spectra [13].

Machine learning techniques such as Gaussian Process Regression (GPR) and LSTM networks have been applied to analyse EIS data for various applications. GPR models have been effectively used for predicting capacitor degradation in power electronics, demonstrating robust performance in forecasting remaining useful life (RUL) based on impedance changes over time [14]. Meanwhile, LSTM networks have been used for MOSFET degradation prediction, leveraging time-series impedance data to identify early signs of component failure. These AI-driven models significantly outperform traditional curve-fitting approaches in terms of adaptability and predictive accuracy.

The integration of AI in EIS analysis has also extended to biomedical applications, where neural networks have been used to classify bacterial concentrations and detect cancerous cells based on impedance measurements. Such approaches provide a non-invasive, rapid diagnostic tool, reducing the reliance on complex laboratory procedures.

Despite these advancements, challenges remain in AI-driven EIS analysis. The accuracy and generalizability of machine learning models heavily depend on the quality and quantity of training data. Additionally, computational costs associated with training deep learning models can be substantial, necessitating the development of efficient algorithms that balance performance and resource utilization. Future research should focus on improving model robustness, exploring transfer learning techniques, and integrating AI with portable EIS solutions for real-time diagnostics.

2.1.3 Predictive Maintenance Using EIS

Predictive maintenance has gained prominence as an effective strategy for minimizing unplanned downtime and extending the lifespan of power electronic components. By leveraging EIS data and AI-driven predictive models, researchers have developed techniques to estimate the RUL of critical components, such as lithium-ion batteries, capacitors, and MOSFETs, enabling timely maintenance interventions.

A notable approach in predictive maintenance involves using time-series EIS data to monitor component degradation and predict failure points. Researchers have employed LSTM neural networks to analyse MOSFET thermal overstress aging data, demonstrating a high R^2 score of 0.7915 in RUL predictions. Additionally, an incremental learning technique was integrated into LSTM models, allowing them to adapt to new data and improve predictive accuracy over time, significantly enhancing maintenance planning for power electronics.

For capacitor degradation prediction, GPR models have been applied to EIS measurements, achieving reliable forecasts of capacitance decay under different voltage stress conditions [14]. These predictive models enable proactive scheduling of maintenance tasks, reducing the risk of unexpected failures and improving system reliability.

In the context of energy storage systems, predictive maintenance using EIS has been explored for lithium-ion battery health monitoring. A cloud-based degradation diagnosis framework was introduced, integrating impedance modelling and AI to assess battery aging parameters in real-time. This approach provided valuable insights into battery degradation modes and facilitated optimized charging strategies to extend battery lifespan [12].

The application of predictive maintenance using EIS has also been extended to fuel cell systems. By analysing impedance spectra, researchers have developed diagnostic techniques to detect drying and flooding phenomena in fuel cell stacks, which are critical for ensuring consistent performance. Such diagnostic techniques, when integrated with AI models, can provide early warnings of operational issues and recommend optimal maintenance schedules [11].

While predictive maintenance using EIS has demonstrated substantial benefits, its practical implementation faces several challenges. The accuracy of RUL predictions depends on the reliability of the collected data, environmental factors, and the complexity of degradation mechanisms. Additionally, real-time deployment of predictive models requires computational efficiency and seamless integration with existing monitoring systems. Future research should focus on refining AI-driven predictive maintenance frameworks, improving data acquisition methods, and developing edge-computing solutions for on-site analysis.

2.2 Limitations of the Previous Studies

The use of EIS for predictive maintenance has gained significant traction in battery diagnostics, fuel cell monitoring, corrosion detection, and industrial equipment maintenance. However, several limitations need to be addressed before EIS-based predictive maintenance systems can be fully adopted in commercial and industrial settings.

One major limitation is the complexity of impedance-based degradation modelling. While EIS provides a wealth of information about battery health, corrosion rates, and material degradation, interpreting these impedance spectra to predict RUL and failure modes is highly challenging [9]. Many predictive maintenance approaches rely on historical degradation data and statistical models, which may not accurately capture real-time operational conditions and dynamic degradation mechanisms [7], [14].

Another issue is the requirement for frequent calibration and baseline measurements. Unlike temperature or vibration sensors commonly used in industrial predictive maintenance, EIS measurements require a stable reference condition to compare impedance changes over time. Variations in operating temperature, electrode condition, electrolyte composition, or external noise can significantly impact impedance readings, leading to false positives or inaccurate failure predictions [12]. This limitation necessitates the development of robust compensation algorithms and adaptive thresholding techniques to enhance the reliability of EIS-based predictive maintenance systems.

Additionally, integration challenges with existing industrial systems present a barrier to large-scale adoption. Most industrial predictive maintenance frameworks are based on machine learning models trained on time-series sensor data such as vibration, temperature, current-voltage characteristics. Integrating EIS as an additional diagnostic tool requires customized hardware, software interfaces, and data fusion strategies, which increases cost and complexity for industries. Moreover, real-time impedance monitoring requires continuous electrical excitation of the system, which may not be feasible in some applications such as battery-powered devices or sensitive electronic systems.

Finally, the computational demands of predictive analytics using EIS data pose another challenge. Predictive maintenance approaches that leverage machine learning models, neural networks, and Bayesian inference methods require high processing power and extensive training data to achieve reliable degradation predictions. However, many low-power industrial devices and embedded systems cannot accommodate such computationally intensive models, necessitating lightweight predictive algorithms optimized for real-time execution.

2.3 Proposed Solutions

To overcome the limitations identified in portable EIS solutions, AI integration in EIS analysis, and predictive maintenance using EIS, several innovative approaches can be implemented. These solutions aim to enhance measurement accuracy, improve AI model robustness, optimize real-time processing, and integrate EIS seamlessly into predictive maintenance frameworks. By addressing these challenges, EIS technology can be effectively utilized for real-time diagnostics, industrial applications, and machine learning-based predictive maintenance.

One of the key limitations of portable EIS devices is their compromised accuracy due to low-cost hardware and limited frequency range. To improve the precision and reliability of portable EIS solutions, several hardware and software optimizations can be introduced. Advanced signal processing and noise reduction techniques, such as adaptive filtering algorithms like Kalman filters and wavelet denoising, can be integrated to eliminate environmental noise and measurement artifacts [7]. This will

significantly improve measurement stability, particularly in biomedical and industrial applications where external factors like electromagnetic interference and temperature fluctuations affect readings. Additionally, the use of multi-frequency measurement protocols can enhance the resolution of impedance spectra, allowing for better material characterization and detection of subtle changes in electrochemical properties.

To enhance AI integration in EIS analysis, a major focus should be placed on improving the training datasets and model generalization capabilities. One of the primary reasons for AI model inaccuracies in EIS analysis is the insufficient amount of labelled impedance spectra for training. Addressing this challenge requires the creation of large, high-quality datasets covering a wide range of electrochemical conditions. Techniques such as data augmentation and transfer learning can also be utilized to compensate for the lack of extensive labelled data. Transfer learning allows AI models to leverage pre-trained features from related tasks, reducing the need for large datasets while improving model accuracy [9].

Real-time AI processing is another challenge in EIS applications due to the computational complexity of impedance data analysis. To address this issue, edge computing and optimized AI inference techniques can be integrated into EIS devices. Instead of relying on cloud-based processing, AI models can be deployed on embedded hardware such as field-programmable gate arrays (FPGAs) or Tensor Processing Units (TPUs) to achieve real-time analysis with minimal latency [11]. Additionally, lightweight neural network architectures, such as MobileNet or TinyML, can be used to reduce computational overhead while maintaining high accuracy. These improvements will enable EIS systems to perform real-time diagnostics in remote or resource-limited settings without requiring constant internet connectivity.

For predictive maintenance using EIS, an effective strategy is the integration of hybrid models that combine physics-based modeling with machine learning. Many existing predictive maintenance models rely solely on historical data, making them less effective in handling real-time system variations. By incorporating physics-informed AI models, which combine electrochemical impedance simulations with real-time data, predictive maintenance frameworks can achieve higher accuracy in estimating component

degradation and failure probabilities [12]. This hybrid approach enables models to learn the underlying electrochemical mechanisms governing battery or material degradation, rather than relying purely on statistical correlations.

Lastly, the implementation of cloud-based predictive maintenance platforms can enable large-scale deployment of EIS-based diagnostics. By integrating IoT-enabled EIS devices with cloud computing infrastructure, real-time impedance measurements can be continuously uploaded and analyzed using AI-driven predictive models. Cloud platforms allow for centralized data storage, cross-device comparison, and automated alert systems, ensuring that maintenance teams receive timely notifications of potential failures. Furthermore, predictive maintenance platforms can leverage federated learning, where AI models are trained collaboratively across multiple decentralized devices without transferring sensitive data to a central server, thereby preserving data privacy and security.

CHAPTER 3

System Methodology/ Approach

In this chapter, we present the design of the portable electrochemical impedance spectroscopy (EIS) measurement device, which is designed to assess the SoC of LiBs.

3.1 System Design Diagram

3.1.1 System Architecture Diagram

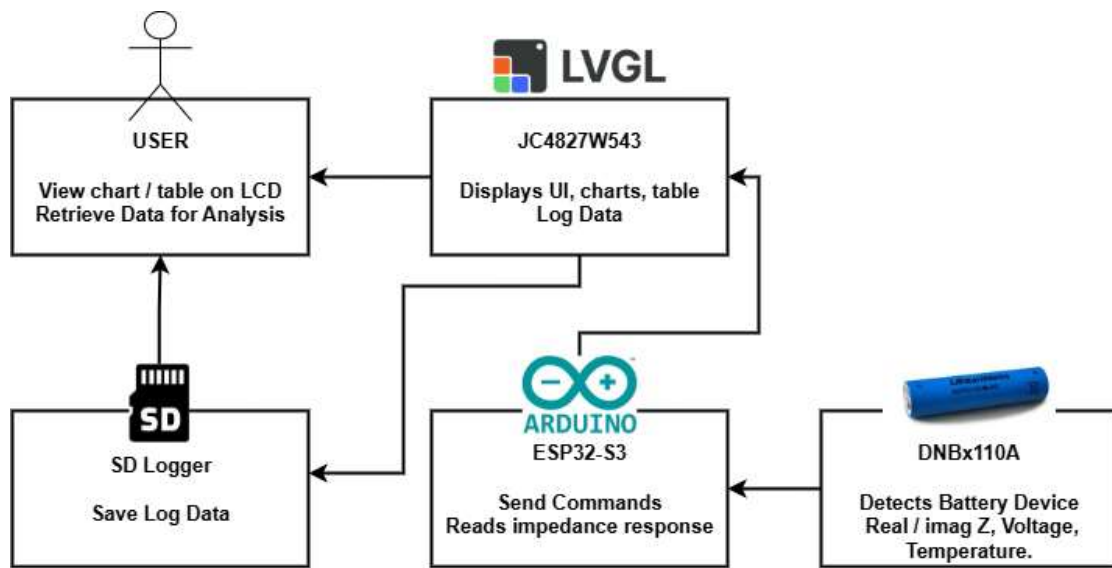


Figure 3.1: System Architecture Diagram for the System

This system architecture illustrates a portable Electrochemical Impedance Spectroscopy (EIS) measurement device. At the core, the ESP32-S3 microcontroller communicates with the DNBx110A module, which detects the battery's real and imaginary impedance values along with voltage and temperature. The ESP32-S3 sends commands and collects responses, then forwards the results to the JC4827W543 LCD display running on LVGL, where the user can view the charts, table and log data in real time. To ensure data persistence, measurement results are also saved onto a SD card, which allows user to retrieve CSV files later for analysis. The user interacts with the system through the LCD interface, gaining access to visualized results and stored logs for further evaluation.

3.1.2 User Case Diagram and Description

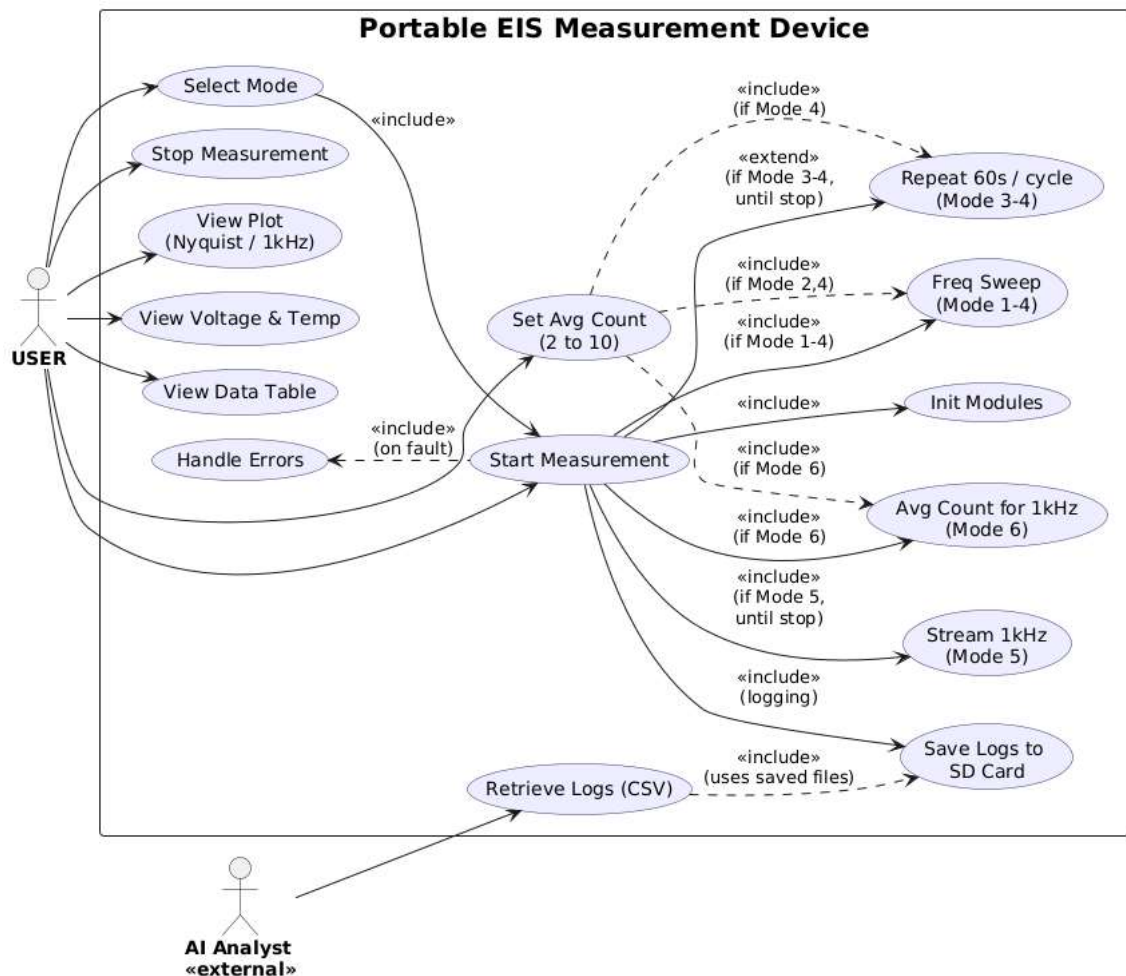


Figure 3.2: User Case Diagram for the System

Based on the use case diagram above, the main interaction between User and the system, as well as the external role of the AI Analyst. The User operates the device by selecting the measurement mode from Mode 1 to Mode 6. Besides that, the system allows User to set the averaging count between 2 to 10, as well as to start or stop measurement. Once started, the system performs different actions depending on the selected mode. For example, it can initialize modules for communication, perform frequency sweeps (Mode 1 – 4), or stream continuous 1kHz samples (Mode 5). In Mode 6, the system uses averaging count chosen by User to perform 1kHz measurements, whereas Mode 3 and Mode 4 also extend functionality with a repeated 60-second wait cycle until the stop command is given. During the process, the User can monitor results through plots (Nyquist or 1kHz), view voltage and temperature readings, and access

tabulated measurement data, particularly in Modes 5 and 6. Lastly, the system also handles errors automatically, ensuring faults are detected and displayed properly.

In addition, the system is designed for persistent data storage and post-analysis. All results are logged and saved automatically onto the SD card in CSV format, which can be accessed later. The User can retrieve these logs for review, while the AI Analyst uses the CSV files for advanced evaluation such as determining the SOH of the battery through machine learning model development. This separation of roles enables the device to be practical for engineers who need live testing feedback, while also supporting researchers who require long-term data analysis for battery characterization and performance insights.

3.1.3 Activity Diagram

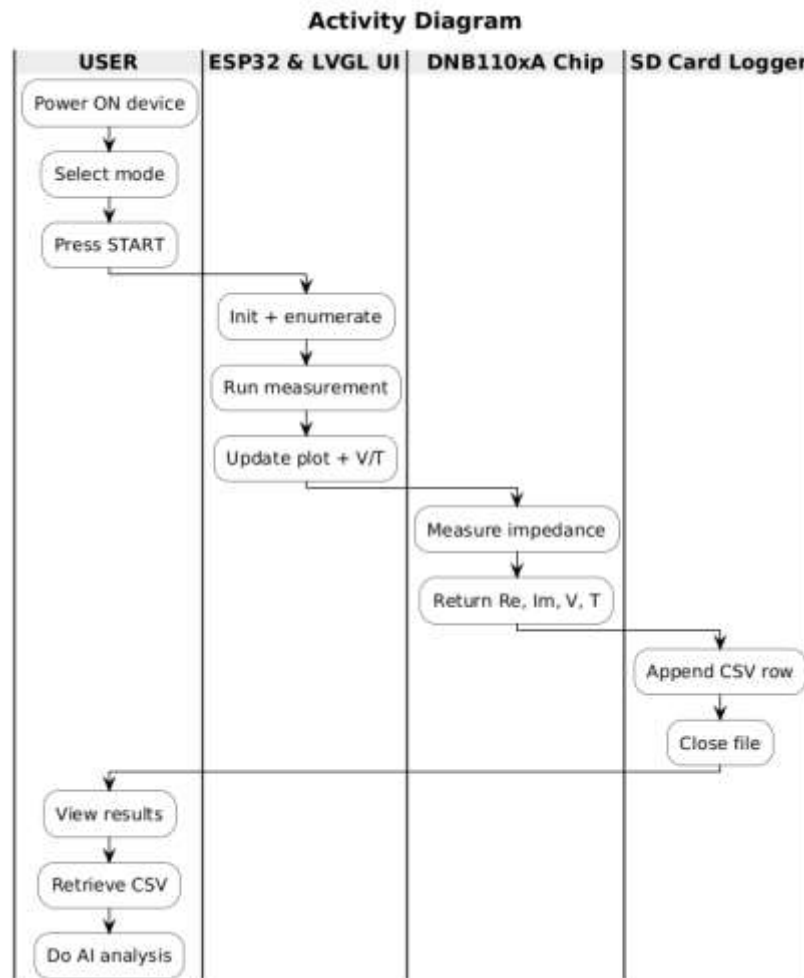


Figure 3.3: Activity Diagram of the System

CHAPTER 4

System Design

In this chapter, we focused on assessing the feasibility of extracting and processing impedance data from LIBs to estimate key parameters like the SoC and SoH. To do this, we used the impedance measurement device (DNB110xA) to acquire the real-time data across a set of predefined frequencies. The process involved extracting the mantissa and exponent from the raw impedance data, followed by applying the corresponding formulas to convert this data into usable real and imaginary parts of the impedance.

Moreover, we designed and developed a GUI using the LVGL (LittleGL) library on the MCXN947 microcontroller to visualize the impedance data. The GUI was designed to display real-time graphs generated from the impedance data captured during the measurement.

4.1 System Block Diagram

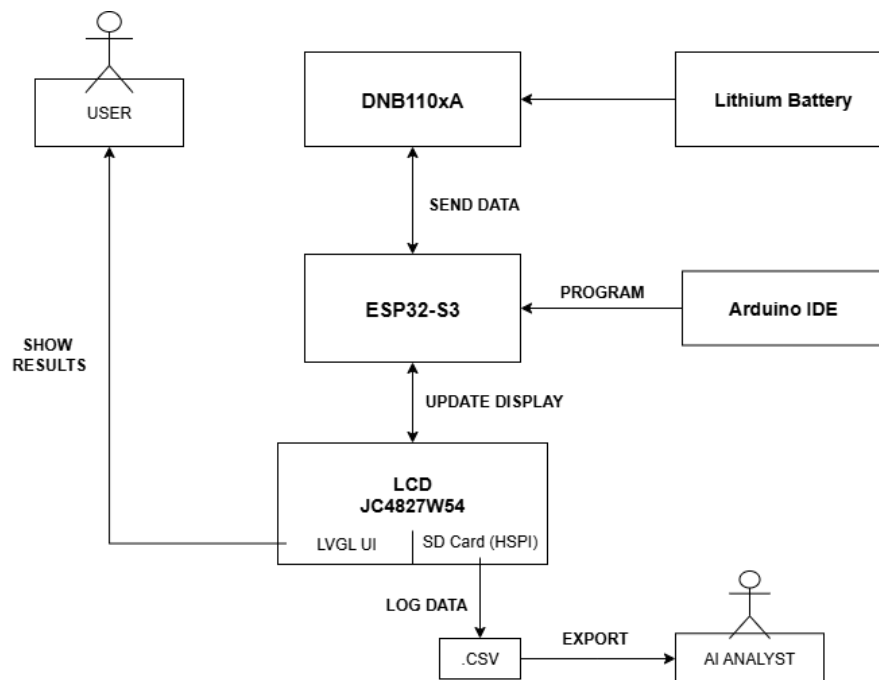


Figure 4.1: System Block Diagram

This block diagram illustrates the overall workflow and data flow of the portable EIS measurement device. The process begins with the lithium battery, which is connected to the DNB110xA impedance measurement IC. The DNB110xA collects impedance, voltage, and temperature values from the battery and sends data to the ESP32-S3 microcontroller. The ESP32-S3, which is programmed through the Arduino IDE using the ESP32S3 DEV MODULE configuration, processes the incoming measurement data and manages both real-time operations and user interaction. The processed results are then transmitted to the LCD JC4827W543, where the LVGL-based user interface updates the display with charts, tables, and voltage and temperature readings, enabling the user to conveniently view results.

Meanwhile, the ESP32-S3 coordinates with the LCD module's integrated SD card logger via HSPI to log data in CSV format. These stored files can be retrieved and exported to an AI Analyst, who performs offline data analysis such as machine learning based SOH prediction and deeper battery performance evaluation.

4.2 System Components Specifications

4.2.1 Hardware

The hardware of the portable EIS Measurement Device is composed of five primary subsystems: the ESP32-S3-WROOM-1 module, which serves as the main processing unit for data handling and communication; the Guition JC4827W543 HMI display, which provides real-time visualization and user interaction through a 4.3-inch capacitive touch interface; the DNB110xA impedance measurement module, which performs electrochemical impedance analysis; the 18650 lithium-ion battery, which acts as the test subject; and the SD card interface, which ensures persistent data logging. Together, these subsystems form a compact and portable platform capable of accurate impedance measurements, efficient data processing, interactive monitoring, and reliable storage for further analysis.

Interface Board

The interface board serves as the interfacing and signal management unit between the DNB110xA impedance measurement IC and the JC4827W543 HMI display. It hosts the ESP32-S3 development module, which is responsible for acquiring impedance data from the DNB110xA, processing it, and transmitting it to the LCD display for visualization.

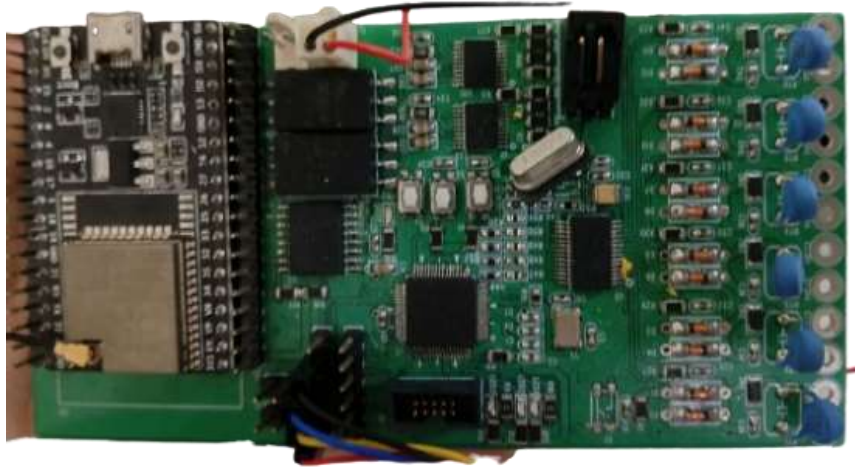


Figure 4.1: ESP32-S3 Interface Board Between DNB110xA and JC4827W543

Table 4.1: Specification of Interface Board

Description	Specification
Processor Module	ESP32-S3 development module (dual-core Xtensa LX7, 240 MHz)
Interfaces Supported	UART (to DNB110xA), SPI/HSPI (to JC4827W543, SD card), GPIO expansion
Components Integrated	Oscillator, signal conditioning ICs, connectors, passive filters
Function	Middle board linking impedance measurement IC with HMI display
Power Supply	3.3V / 5V compatible (regulated from main system)

Impedance Measurement Controller (DNB110xA)

The DNB110xA is an impedance measurement integrated single-cell controller IC used for analyzing the electrochemical impedance of lithium-ion batteries. It plays a crucial role in obtaining accurate impedance data for battery diagnostics.



Figure 4.3: DNB110xA

Table 4.2: Specification of DNB110xA

Description	Specification
Model	DNB110xA
Function	Electrochemical Impedance Spectroscopy (EIS) Measurement IC
Measurement Accuracy	High precision impedance analysis
UART Pins (ESP32-S3)	UART Pins (ESP32-S3)
Communication Interfaces	SPI/I2C for MCU communication
Operating Voltage	3.3V/5V compatible
Power Consumption	Low-power operation mode

LCD Display - Guition JC4827W543 HMI

The Guition JC4827W543 HMI display functions as both the main controller and user interface of the Portable EIS Measurement Device, integrating an ESP32-S3-WROOM-1 module directly on the LCD board. This module combines dual-core processing (up to 240 MHz), 512 KB SRAM, and 8 MB PSRAM to handle signal processing, UI rendering with LVGL 9.3, and data logging. Its 4.3-inch IPS LCD (480×272 pixels) with a GT911 capacitive touch controller provides an interactive interface, while built-in Wi-Fi, Bluetooth, and an SD card slot enable IoT connectivity and CSV data storage, making it a fully self-contained HMI system.



Figure 4.4: 4.3-inch LCD Display

Table 4.3: Specification of JC4827W543 HMI Display with Integrated ESP32-S3-WROOM-1

Description	Specification
Model	Guition JC4827W543 HMI
ESP32-S3-WROOM-1 Module	Dual-core Xtensa® LX7 SoC, up to 240 MHz, Wi-Fi 802.11 b/g/n, Bluetooth 5.0 LE
Resolution	4.3-inch IPS TFT LCD, 480×272 pixels resolution
Interface	Parallel RGB connection to ESP32-S3
Touch Controller	GT911 capacitive touch (I2C)
Touch Pins	SDA = 8, SCL = 4, INT = 3, RST = 38

Memory	512 KB SRAM, 8 MB PSRAM
Data Storage	Integrated SD card slot (FAT32 logging in CSV format)
Features	Runs LVGL 9.3 UI for charts, tables, and battery health visualization
Programming Interface	USB-C connector

18650 Lithium Battery

The test subject for impedance measurement is a standard 18650 lithium-ion battery, widely used in consumer electronics. It is directly connected to the DNB110A module for SOH and performance evaluation.



Figure 4.5: 18650 Lithium Battery Cell

Table 4.4: Specification of 18650 Lithium Battery

Description	Specification
Type	Rechargeable lithium-ion cell
Standard Voltage	3.7 V nominal
Capacity	2000–3500 mAh (depending on manufacturer)
Form Factor	18 mm diameter × 65 mm length
Connection	Direct interface to DNB110A measurement terminals

SD Card

The SD card module provides persistent data storage for measurement results. Connected to the ESP32-S3 through the HSPI interface, it stores impedance, voltage, and temperature data in CSV format. The dual-core configuration of the ESP32-S3 ensures efficient operation by assigning measurement tasks to Core 1 and logging tasks to Core 0, allowing parallel execution without interference.



Figure 4.6: SD Card Module for Data Logging

Table 4.5: Specification of LCD Display

Description	Specification
Model	SD Card Module
Interface	HSPI (High-Speed SPI)
SPI Pins (ESP32-S3)	MISO = 13, MOSI = 11, SCLK = 12, CS = 10
Supported Formats	FAT32 file system
Logging Function	CSV data logging with organized folder structure
Operation	Core 0 handles logging

4.2.2 Software

Arduino IDE



Figure 4.7: Arduino Logo

Arduino IDE served as the primary development environment for the ESP32-S3 firmware. It provided a simple toolchain for code editing, library management, compilation, and USB flashing, with board options configured for the ESP32-S3 Dev Module (240 MHz CPU, PSRAM enabled, large partition for assets and logs). Within this IDE, the project integrated core libraries (e.g., SPI, SD, HardwareSerial) to handle the SD logger, high-speed UART at 1 Mbaud to the DNB110xA, and tasking constructs for smooth background logging. The consistent build system and serial monitor accelerated debugging of initialization, device enumeration, and measurement modes, helping the team iterate quickly and keep the firmware stable.

LVGL v9.3



Figure 4.8: LVGL Logo

LVGL v9.3 powered the entire on-device interface. It rendered the Nyquist plot, the 1 kHz traces, voltage and temperature readouts, status dialogs, and mode selectors, all with smooth touch control through the GT911 controller. The project used LVGL widgets, styling, and the draw engine to build responsive screens with overlay dialogs for progress, errors, and countdowns. Custom assets such as a Malaysia flag image and a compact font that includes the Omega symbol were converted to C arrays and compiled into the firmware for fast offline rendering. LVGL events and layout utilities kept the UI responsive while measurements and SD writes ran in parallel, giving the device a polished and appliance-like experience.

4.3 Circuits and Components Design

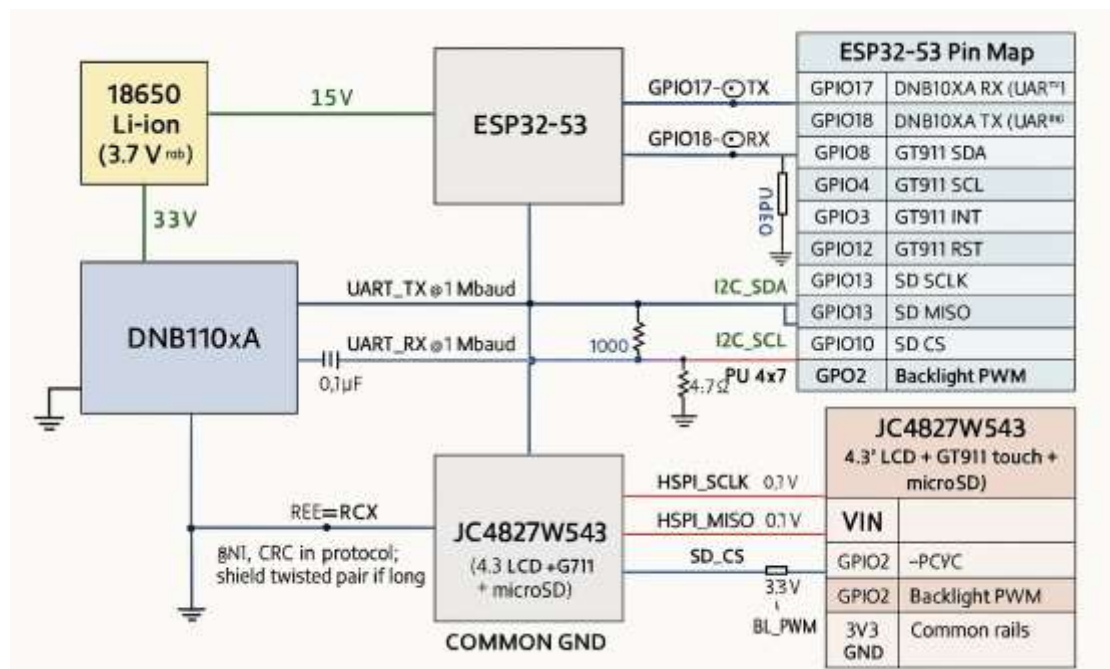


Figure 4.9: Circuit and pin-level interconnect of the portable EIS device.

Figure 4.9 shows the complete interconnect for the portable EIS device. The 18650 cell feeds the DNB110xA impedance front end, which excites the cell and returns complex impedance data. An ESP32-S3 supervises the measurement over a high-speed UART link to the DNB110xA, then updates the JC4827W543 touch display via RGB lines while reading touch coordinates from the GT911 over I²C and driving the backlight with a PWM pin. A dedicated HSPI bus writes CSV logs to the microSD card. This topology isolates the measurement path from the display and storage buses, so the interface stays responsive while acquisition and logging run in parallel.

i'gu're

4.4 System Components Interaction Operation

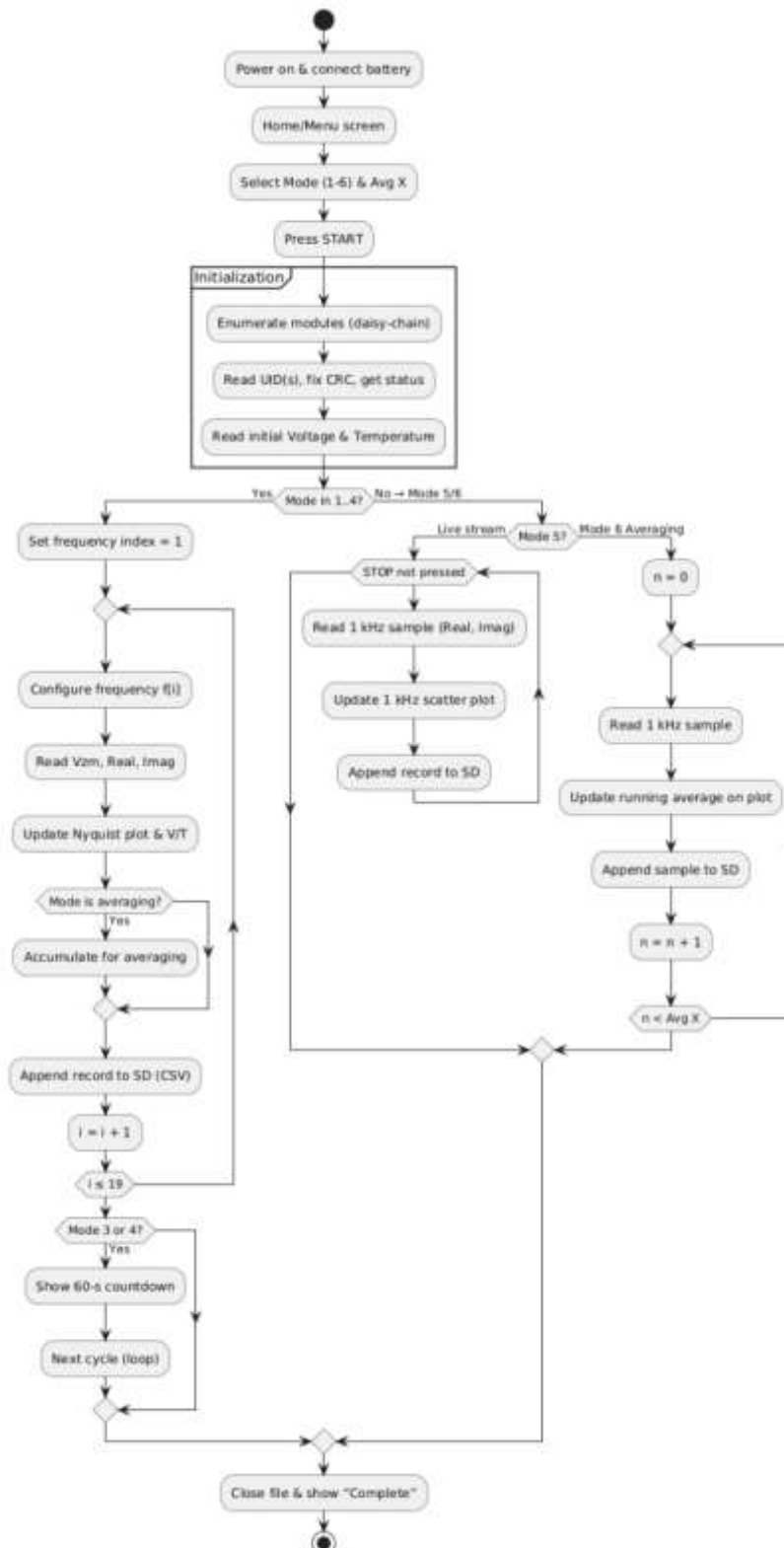


Figure 4.10 : Flowchart of the System

Referring to the flowchart of the system, the operation started when the user selected a measurement mode and an averaging count on the LCD touchscreen, after which the ESP32-S3 initialized the DNB110xA by enumerating modules, verifying UID and CRC, and capturing baseline voltage and temperature. For Modes 1 to 4 the device swept nineteen preset frequencies, updated the Nyquist and voltage–temperature displays, and logged each measurement to the SD card, while Modes 3 and 4 also included a 60-second rest between cycles. For Mode 5 the device continuously acquired 1 kHz samples for real-time plotting and logging, and for Mode 6 it collected 1 kHz samples until the selected average was reached, maintaining a running on-screen average and storing the results. Every path ended with a safe closure of the CSV file and a clear Complete status for the user, after which logs were available for offline analysis.

CHAPTER 5

System Implementation

5.1 Hardware Setup

The Portable EIS Measurement Device was assembled and wired prior to testing. The ESP32-S3 on the interface board acted as the controller for the DNB110xA impedance module and the JC4827W543 4.3" touch display, while a single 18650 Li-ion cell served as the test subject. The LCD board's microSD slot was used for CSV logging. A common ground was established across all modules, and signal runs were kept short to maintain integrity at high baud rates.

Connected components (what was installed):

- **Battery fixture & test cell (18650):**
Only the inserted cell was active/usable, and it was wired with short, polarized leads to the DNB110xA measurement terminals.
- **DNB110xA impedance module:**
Performed EIS and communicated with the ESP32-S3 over high-speed UART for command/response and data streaming.
- **ESP32-S3 interface board (middle board):**
Served as the central controller that handled initialization, data acquisition, UI updates, and SD logging.
- **JC4827W543 LCD touch display (GT911):**
Ran the LVGL UI to show Nyquist plots, 1 kHz traces, tabulated data, voltage, temperature, and status; the backlight was driven by PWM.
- **microSD storage (on LCD board):**
Logged each record in FAT32/CSV format for later analysis.

Power and grounding

- The LCD/HMI was powered via USB-C 5 V; onboard regulators supplied the 3.3 V rails.
- Common GND was shared by the ESP32-S3, DNB110xA, LCD, and battery negative.



Figure 5.1 : Hardware Setup. Battery fixture, ESP32-S3 interface board with DNB110xA, and JC4827W543 display showing the LVGL “EIS Battery Monitor” screen

Table 5.1 Pin Connection of ESP32-S3 and DNB110Xa

ESP32-S3 Pin	DNB110xA
GPIO17	RX
GPIO18	TX
GND	GND

Table 5.2 Pin Connection of ESP32-S3 and GT911 Touch Driver

ESP32-S3 Pin	GT911
GPIO8	SDA
GPIO4	SCL
GPIO3	INT
GPIO38	RST
GND	GND

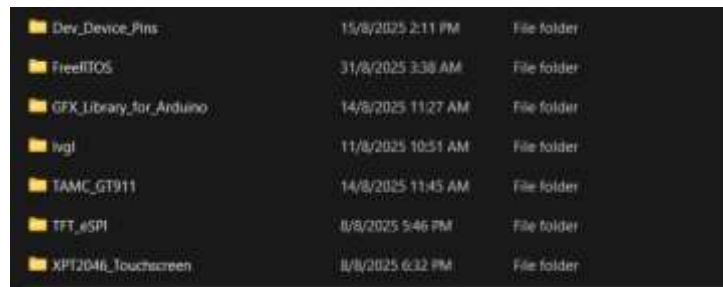
Table 5.3 Pin Connection of ESP32-S3 and microSD (HSPI)

ESP32-S3 Pin	GT911
GPIO12	SCLK
GPIO11	MOSI
GPIO13	MISO
GPIO10	CS
GND	GND

5.2 Software Setup

5.2.1 Arduino Libraries Used

The firmware was developed in the Arduino IDE for the ESP32-S3 target. The following libraries were installed and configured to support display rendering, touch input, storage, and scheduling.



Dev_Device_Pins	15/8/2025 2:11 PM	File folder
FreeRTOS	31/8/2025 3:38 AM	File folder
GFX_Library_for_Arduino	14/8/2025 11:27 AM	File folder
lvgl	11/8/2025 10:51 AM	File folder
TAMC_GT911	14/8/2025 11:45 AM	File folder
TFT_eSPI	8/8/2025 5:46 PM	File folder
XPT2046_Touchscreen	8/8/2025 6:32 PM	File folder

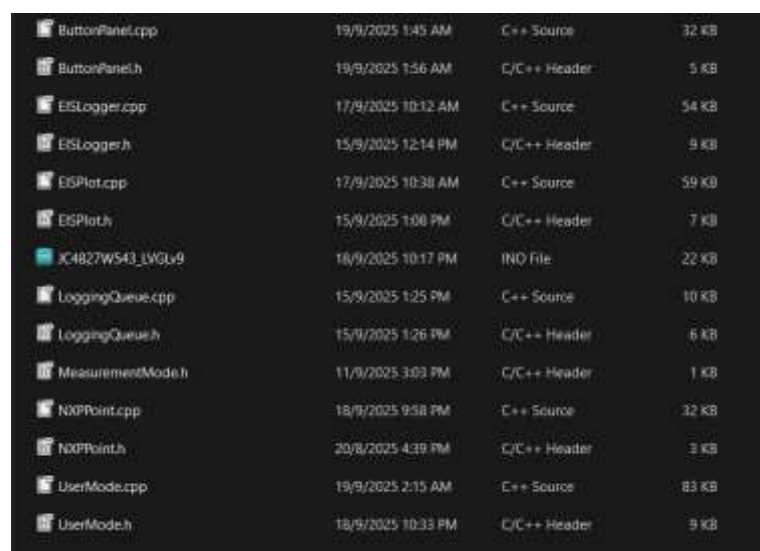
Figure 5.2: Libraries Used

- ESP32 Board Core (Espressif)**
 Provided device drivers and HAL for UART, SPI/HSPI, I²C, timers, PSRAM, and FreeRTOS on the ESP32-S3.
- FreeRTOS**
 Separated UI rendering from logging. A producer task pushed samples into a queue and a logger task handled SD writes to prevent UI stalls.
- LVGL 9.3**
 Drove the graphical interface and charts. It supported double buffering, event handling, and a modular widget set suitable for real-time plots and tables.
- TAMC_GT911**
 Supplied the I²C driver for the GT911 capacitive touch controller on the JC4827W543, which fed LVGL with touch events.
- SPI and SD**
 Enabled HSPI communication and FAT32 file operations for CSV logging on the microSD card at a stable 10 MHz bus speed.
- TFT_eSPI and GFX_Library_for_Arduino**
 Assisted with early bring-up and provided helper routines used by the LVGL display flush function where appropriate.

5.2.2 Source File Organization

To keep the codebase maintainable, functionality was split across modules:

- **JC4827W543_LVGLv9.ino** initialized the board, LVGL, touch, and tasks, and launched the main screen.
- **UserMode.{h,cpp}** implemented the UI flow for mode selection, start and stop actions, and status prompts.
- **MeasurementMode.h** defined mode parameters and frequency sets.
- **NXPPoint.{h,cpp}** framed and parsed the command and response data for the DNB110xA.
- **EISPlot.{h,cpp}** created Nyquist and 1 kHz charts, including decimation for smooth refresh.
- **ButtonPanel.{h,cpp}** packaged reusable LVGL widgets such as mode buttons and the averaging selector.
- **LoggingQueue.{h,cpp}** implemented a FreeRTOS queue and mutex for thread-safe logging.
- **EISLogger.{h,cpp}** mounted the SD card, created session folders, and appended CSV rows.



ButtonPanel.cpp	19/9/2025 1:45 AM	C++ Source	32 KB
ButtonPanel.h	19/9/2025 1:56 AM	C/C++ Header	5 KB
EISLogger.cpp	17/9/2025 10:12 AM	C++ Source	54 KB
EISLogger.h	15/9/2025 12:14 PM	C/C++ Header	9 KB
EISPlot.cpp	17/9/2025 10:38 AM	C++ Source	59 KB
EISPlot.h	15/9/2025 1:06 PM	C/C++ Header	7 KB
JC4827W543_LVGLv9	18/9/2025 10:17 PM	INO File	22 KB
LoggingQueue.cpp	15/9/2025 1:25 PM	C++ Source	10 KB
LoggingQueue.h	15/9/2025 1:26 PM	C/C++ Header	6 KB
MeasurementMode.h	11/9/2025 3:03 PM	C/C++ Header	1 KB
NXPPoint.cpp	18/9/2025 9:58 PM	C++ Source	32 KB
NXPPoint.h	20/8/2025 4:39 PM	C/C++ Header	3 KB
UserMode.cpp	19/9/2025 2:15 AM	C++ Source	83 KB
UserMode.h	18/9/2025 10:33 PM	C/C++ Header	9 KB

Figure 5.3: Project Source Code

5.2.3 Ohm Symbol integration

The default UI font did not include Greek glyphs, a minimal custom font that contained only the Ω character was generated and added as a C source file. The file was stored under ...\\Arduino\\libraries\\lvgl\\src\\font\\ as **montserrat_10_ohm.c** and **montserrat_12_ohm.c**. The symbol was then referenced in labels by defining a small macro and applying the custom font to the control that required it.

```
// Axis labels
lv_obj_t* x_label = lv_label_create(container);
char buf[64];
snprintf(buf, sizeof(buf), "Real Impedance (m%s)", OHM_SYMBOL);
lv_label_set_text(x_label, buf);
lv_obj_set_style_text_font(x_label, &montserrat_10_ohm, 0);
lv_obj_set_style_text_color(x_label, lv_color_hex(0xFFFFFFFF),
                           LV_PART_MAIN | LV_STATE_DEFAULT);
lv_obj_align_to(x_label, chart_container, LV_ALIGN_OUT_BOTTOM_MID, 0,
5);
```

Figure 5.4: LVGL snippet rendering the Ω symbol

5.2.4 LVGL Image Asset Integration

Project graphics such as the **Malaysia flag** were converted to LVGL-compatible C arrays and included in the build.

Conversion and placement

1. The PNG image was converted using the **LVGL Image Converter** with v9 output.
2. The generated C file was placed
att ...\\Arduino\\libraries\\lvgl\\src\\widgets\\image\\malaysia_flag.c
and included in the project.

```
// Display Malaysia flag
uint16_t scale = (uint16_t)((uint32_t)256 * 62 / 500);
lv_obj_t* flag = lv_image_create(lv_screen_active());
lv_image_set_src(flag, &malaysia_flag);
lv_obj_align(flag, LV_ALIGN_TOP_MID, -210, -130);
lv_image_set_scale(flag, scale);
```

Figure 5.5: LVGL snippet displaying the Malaysia Flag from a converted C-array image

5.2.5 SD Initialization

To support unattended logging and proof of measurement, the microSD card needed a predictable, fault-tolerant bring-up. The firmware mounted the card at boot, verified write access, and only then enabled the logger. This avoided half-written files and allowed the device to keep measuring even when no card was present.

```
spiSD.begin(SD_SCK, SD_MISO, SD_MOSI, SD_CS);
if (SD.begin(SD_CS, spiSD, 10000000)) {
    Serial.println("SD Card initialized for Core 0 logging");

    // Create data directory
    if (!SD.exists("/data")) {
        SD.mkdir("/data");
    }
    logging_enabled = true;
} else {
    Serial.println("SD Card failed - logging disabled");
    logging_enabled = false;
}
```

Figure 5.6: SD initialization on boot showing successful mount at 10 MHz

5.3 Setting and Configuration

The device was configured on the LCD touch screen and in firmware so acquisition, visualization and logging worked coherently. The user selected the Mode (1 – 6) and averaging count, the controller established the 1Mbaud UART link with UID/CRC checks, captured baseline voltage/temperature, and mounted the SD card with a fixed /data path for CSV output. Display options (brightness, grids/axes, Ω units) were adjusted as needed to ensure repeatable measurements.

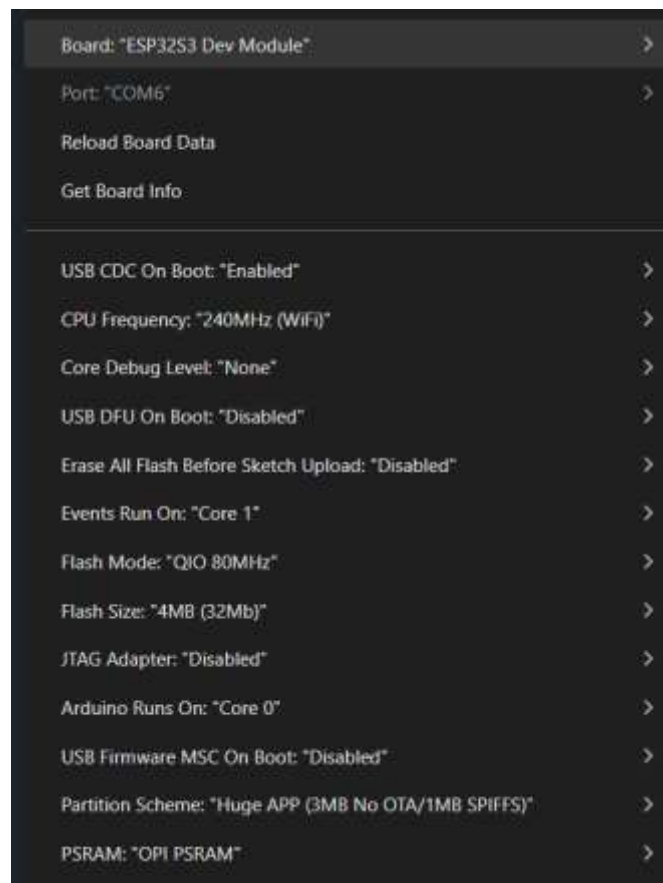


Figure 5.7: Arduino IDE settings for ESP32-S3

The firmware was configured and flashed in the Arduino IDE using the ESP32-S3 Dev Module profile to provide a stable, repeatable runtime for measurement, visualization, and logging. Board options were selected to match the hardware shown in Figure 5.x, which CPU frequency was set to 240 MHz, PSRAM was enabled (OPI PSRAM), and the partition scheme was Huge APP (3MB No OTA / 1 MB SPIFFS) so that LVGL assets and program code could coexist without memory pressure. USB-CDC on boot was

enabled for serial diagnostics, core debug level remained None, and the flash subsystem used QIO @ 80 MHz with a 4 MB device. Runtime placement followed the ESP32-S3 dual-core model in which Arduino tasks ran on Core 0 while event callbacks were scheduled on Core 1. The separation of the tasks allowed the UI and logger to operate smoothly under sustained writes. After flashing, a short bring-up confirmed PSRAM availability and basic peripheral health before the user interface was enabled.



Figure 5.8: EIS Battery Monitor v9.3 UI

Display and touch were initialized at startup using LVGL 9.3 and the GT911 capacitive controller. A minimal calibration step verified that touch coordinates mapped correctly to the 4.3-inch panel so that taps on the chart area and UI buttons were interpreted accurately. The main screen (Figure 5.x) then exposed all user-adjustable parameters. On the right panel, the operator chose among six modes. Modes 1 – 4 performed a 19-frequency sweep that covered (97 Hz up to 4943 Hz). Modes 3 and 4 paused for 50 seconds between cycles to allow stabilization. Mode 5 streamed 1kHz samples in real time. Mode 6 captured 1 kHz samples with averaging. The Avg X stepper underneath let the user select a running mean from 2 up to 10. Start and Stop controlled session execution and were enabled only when the system had passed readiness checks.

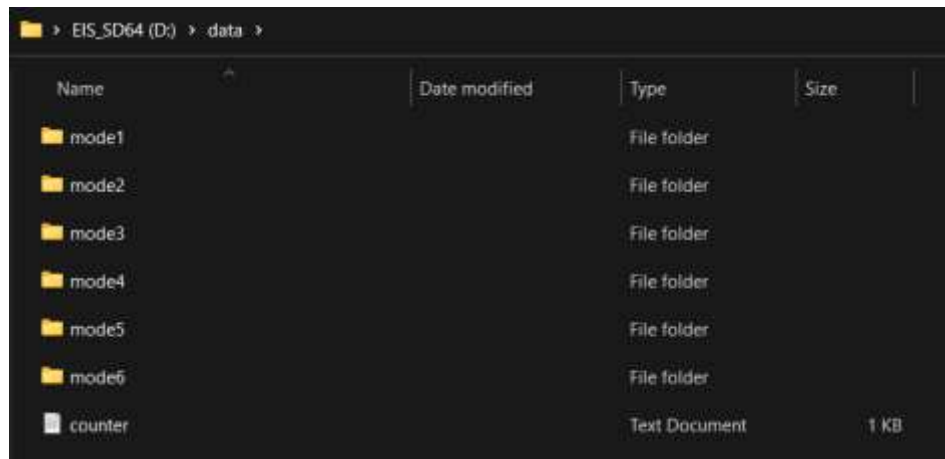


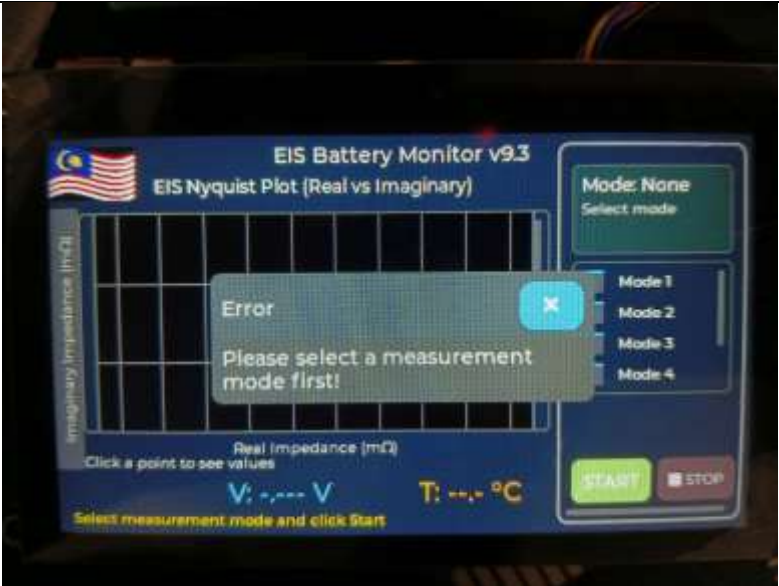
Figure 5.9: SD card File Structure

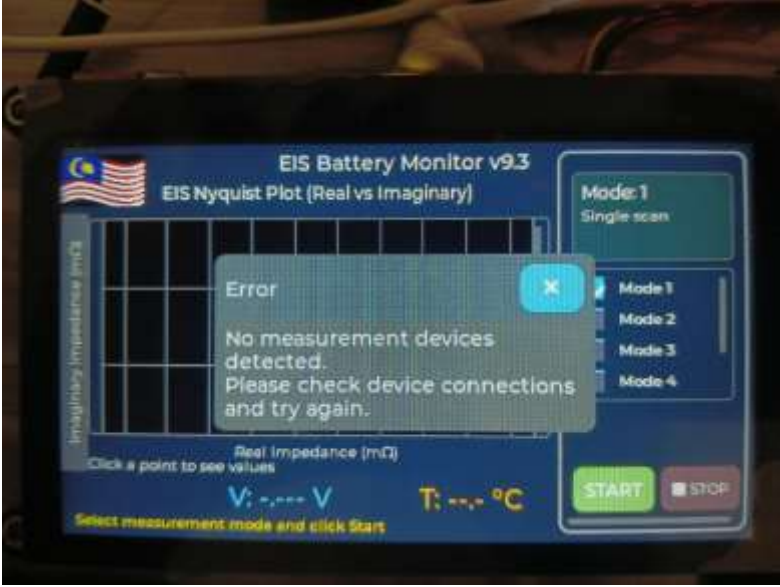
Data logging worked automatically in the background, so the user did not manage storage directly. On boot the firmware mounted a FAT32 microSD card over the HPSI bus and confirmed that a fixed data path existed. The logger organized results by creating folders named mode 1 through mode 6 and then wrote a timestamped file in the active mode folder during each session. In support of compact files and focused analysis, every record contained only the real and imaginary parts of impedance. Voltage and temperature were shown on the screen but intentionally not written to the CSV. A small counter file maintained sequential naming. Logging became active only after a successful mount and directory check, files were always closed safely at completion or when the user pressed Stop.

5.4 System Operation

After configuration the system ran a complete session by verifying modules and then executing either a 19-point sweep (Modes 1–4) or a 1 kHz acquisition (Modes 5–6). It updated plots and tables in real time while appending timestamped records to CSV. Any faults were surfaced immediately and logging was gated to protect files. When the session finished or the user pressed Stop, the log was safely closed and a summary screen was presented for review.

Error handling

No mode selected	<div></div> <p><i>Figure 5.10: Mode not selected warning</i></p> <p>When the user pressed Start without choosing a mode, the interface displayed an error dialog prompting the user to select a measurement mode first. The run did not begin and no file was created. (see Figure 5.xa)</p>
------------------	---

<p>No measurement device detected</p>	 <p><i>Figure 5.11: No device detected warning</i></p> <p>If the DNB110xA was not enumerated or the cell was not connected, the interface showed a clear message indicating that no measurement devices were detected and advised checking the wiring. The system held the Start state until the fault cleared. (see <i>Figure 5.xb</i>)</p>
<p>No SD card inserted</p>	<p>If the SD card was missing or failed to mount, logging was disabled. The UI still allowed viewing live plots, but no CSV file was created, and no data were saved. The status bar reminded the user to insert a card before running another session.</p>

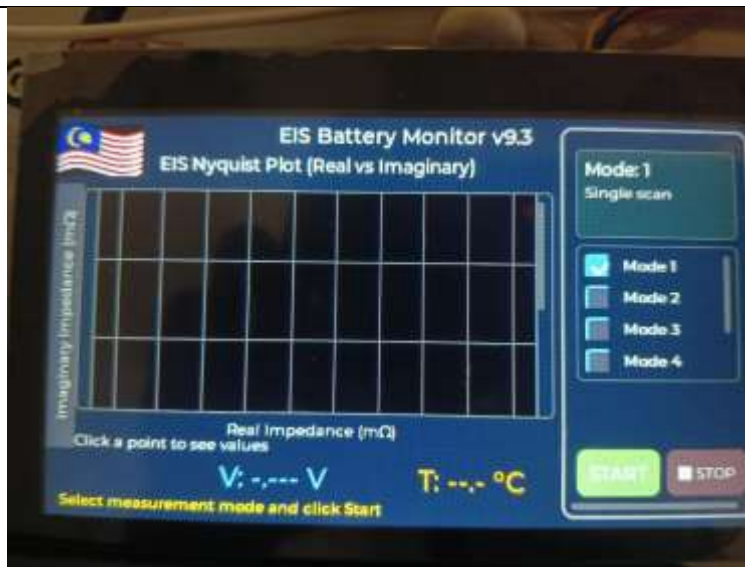
Mode 1**Selection and setup**


Figure 5.12: Mode 1 selection screen

The user selected Mode 1 on the right panel. The header confirmed “Mode 1 Single scan”. The Nyquist canvas and the status strip with live voltage and temperature were shown. (see *Figure 5.xc*)

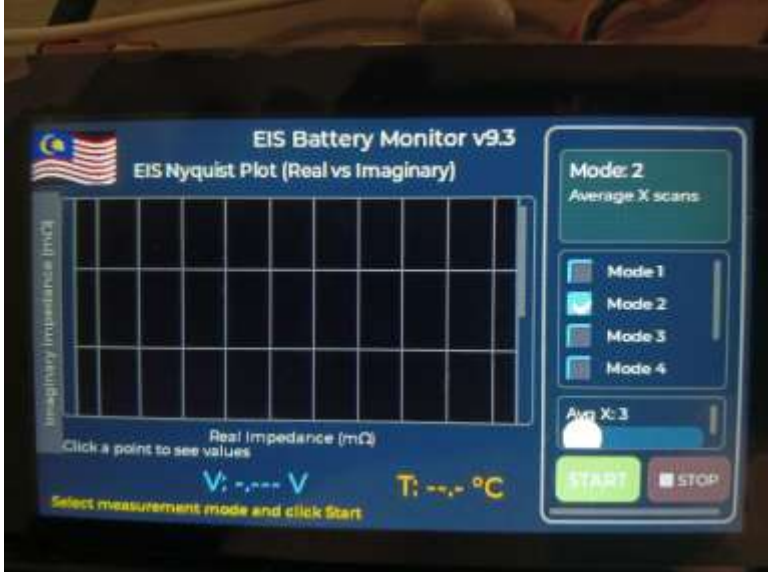

Initialization and progress

Figure 5.13: Sweep progress shown

The system stepped through the predefined list of 19 frequencies from roughly 97 Hz up to 4.943 kHz. For each point it acquired real and imaginary impedance, updated the Nyquist plot, and

	<p>appended a record to the session CSV. The CSV contained only z_real and z_imag for compact storage. Voltage and temperature were shown on screen but not written to the file.</p>
Measurement and logging	 <p><i>Figure 5.14: Sweep complete status</i></p> <p>When the final frequency was measured, the progress panel displayed “Freq 19/19 (100%)” and the status line changed to “Complete”. The logger closed the file safely and the finished sweep remained plotted on the Nyquist canvas for review. (see <i>Figure 5.xe</i> and <i>Figure 5.xf</i>)</p>
Outcome	<p>The user could now select another mode, adjust the averaging count, or start a new session. The saved file was placed automatically under <code>/data/model1</code> with a timestamped name for later analysis.</p>

Mode 2

Selection and setup	 <p><i>Figure 5.15: Mode 2 selection with Avg X</i></p> <p>The user selected Mode 2 on the right panel. The header confirmed “Mode 2 Average X scan”. (see Figure 5.xc)</p>
Averaged acquisition	 <p><i>Figure 5.16: Averaging sweep 1 of X</i></p>

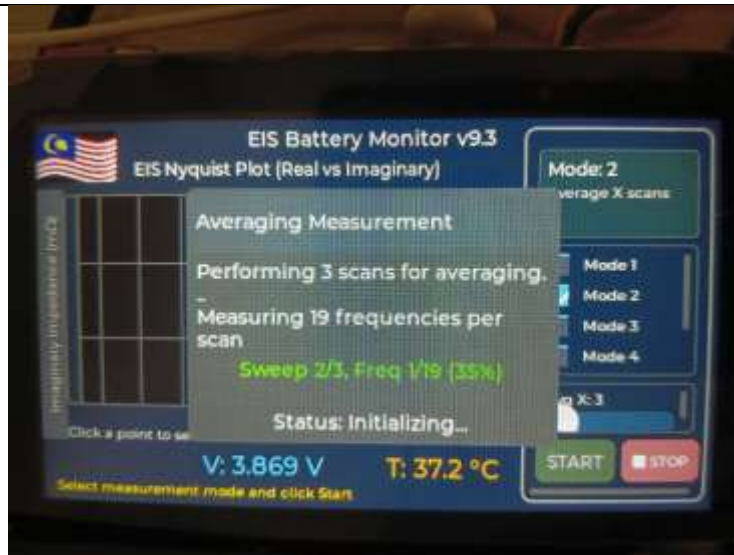


Figure 5.17: Averaging sweep 2 of X



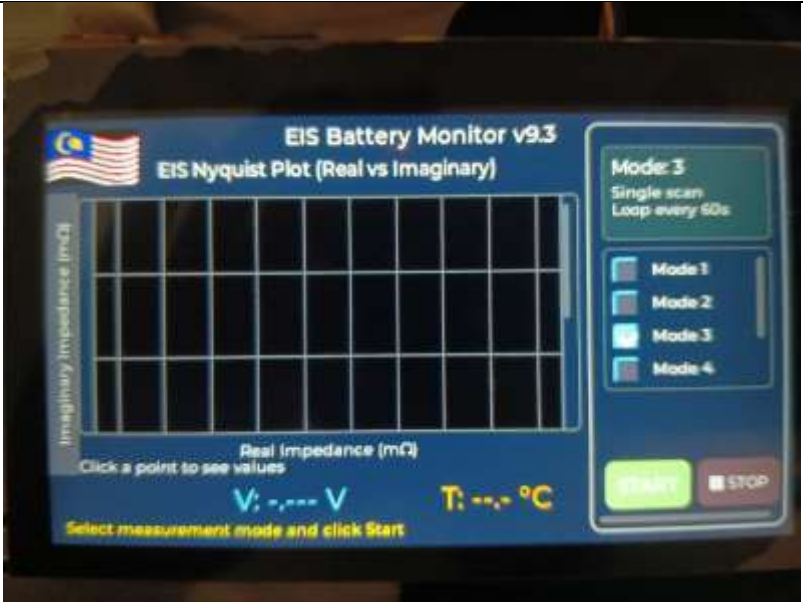

Figure 5.18: Averaging sweep 3 of X



The device executed X complete sweeps of the 19-point frequency set (≈ 97 Hz to 4.943 kHz). For each frequency, it collected one point per sweep and computed an average across the X repeats before updating the Nyquist plot. The progress panel advanced through Sweep 1/X, 2/X, ... and Freq 1/19 \rightarrow 19/19.

Completion	 <p><i>Figure 5.19: Averaging sweep X of X complete</i></p> <p>When the final frequency in the averaged was measured, the dialog Averaging Measurement reported “Sweep 3/3, Freq 19/19 (100%)” and the status line changed to “Complete” as shown. The logger closed the file safely and the finished sweep remained plotted on the Nyquist canvas for review. (see Figure 5.xe and Figure 5.xf)</p>
Results	 <p><i>Figure 5.20: Averaged Nyquist result</i></p>
Outcome	<p>The user could now select another mode, adjust the averaging count, or start a new session. The saved file was placed</p>

	automatically under /data/mode2 with a timestamped name for later analysis.
--	---

Mode 3

Selection and setup	<div></div> <p><i>Figure 5.21: Mode 3 selection screen</i></p> <p>The user selected Mode 3 on the right panel. The header confirmed “Mode 3 Single Scan Loop every 60s”. (see Figure 5.xc)</p>
Initialization	<div></div> <p><i>Figure 5.22: Continuous mode started</i></p>

	<p>A modal dialog appeared: “Continuous measurement started. Scans every 60 seconds. Press STOP to end.” The progress line advanced through Freq 1/19 → 19/19 for the current sweep.</p>
Waiting Interval	 <p><i>Figure 5.23: Waiting between cycles</i></p> <p>After completing the first sweep, the dialog switched to a countdown (e.g., “Waiting 52 seconds...”), indicating the rest period before the next cycle. At the next minute mark, another 19-point sweep began automatically. This loop continued until the user pressed STOP.</p>
Completion	 <p><i>Figure 5.24: Plot after recent cycle</i></p>

	While the plot was displayed, the user could tap any data point to read out the frequency, Real ($m\Omega$), and Imag ($m\Omega$) values in the on-screen readout.
Outcome	The user could now select another mode, adjust the averaging count, or start a new session. The saved file was placed automatically under /data/mode3 with a timestamped name for later analysis.

Mode 4

Selection and setup

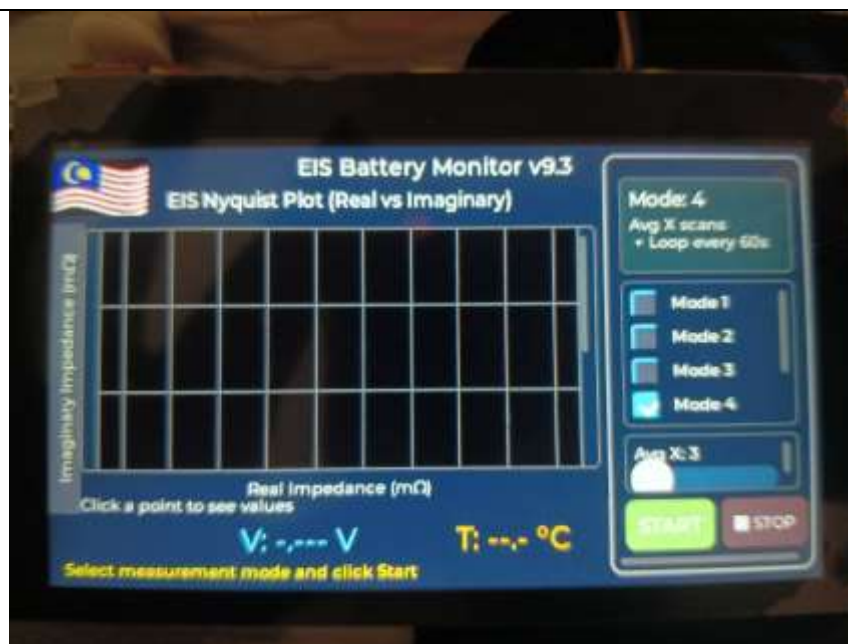
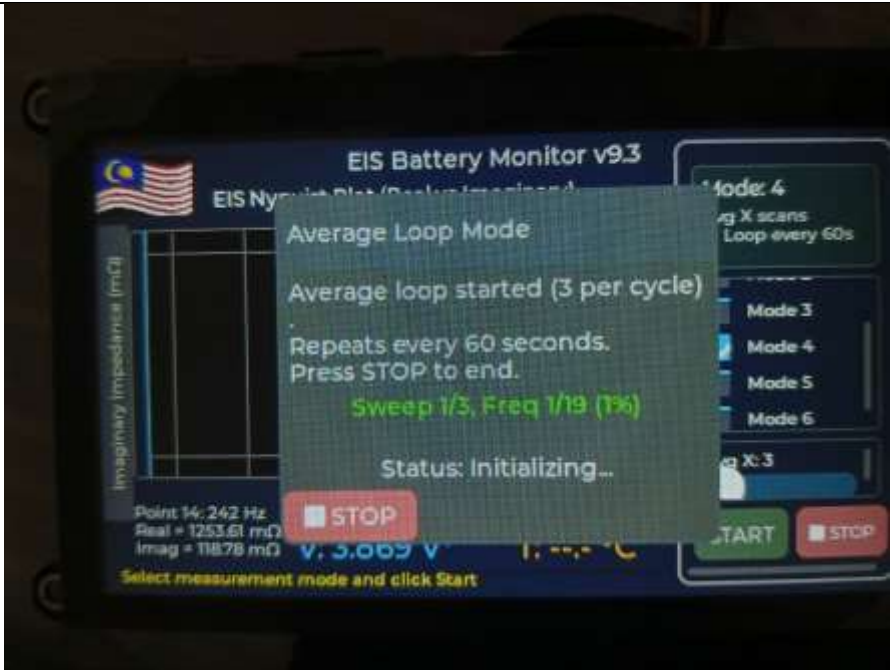



Figure 5.25: Mode 4 selection with Avg X

The user selected Mode 4 on the right panel. The header confirmed “Mode 4 Avg X scan + Loop every 60s”. (see Figure 5.xc)

<p>Averaged cycling</p>	<div data-bbox="491 190 1385 857"><p>The screenshot shows the 'Average Loop Mode' interface. A central text box displays: 'Average loop started (3 per cycle)', 'Repeats every 60 seconds. Press STOP to end.', and 'Sweep 1/3, Freq 1/19 (1%)'. Below this, it says 'Status: Initializing...'. At the bottom of the text box is a red 'STOP' button. The background interface includes a graph of Imaginary Impedance (mΩ) vs. Frequency (Hz), a list of modes (Mode 3 to Mode 6), and a 'START' button. At the bottom, it shows 'Point 14: 242 Hz', 'Real = 1253.61 mΩ', 'Imag = 118.78 mΩ', 'V: 3.065 V', and 'T: 36.0 °C'.</p></div> <p data-bbox="678 880 1200 913"><i>Figure 5.26: Average loop sweep 1 of X</i></p> <div data-bbox="502 929 1369 1579"><p>This screenshot is similar to Figure 5.26 but shows 'Sweep 2/3, Freq 9/19 (49%)'. The 'Status' remains 'Initializing...'. The 'STOP' button is still present. The background interface elements are consistent with the previous figure.</p></div> <p data-bbox="678 1601 1200 1635"><i>Figure 5.27: Average loop sweep 2 of X</i></p>
-------------------------	--

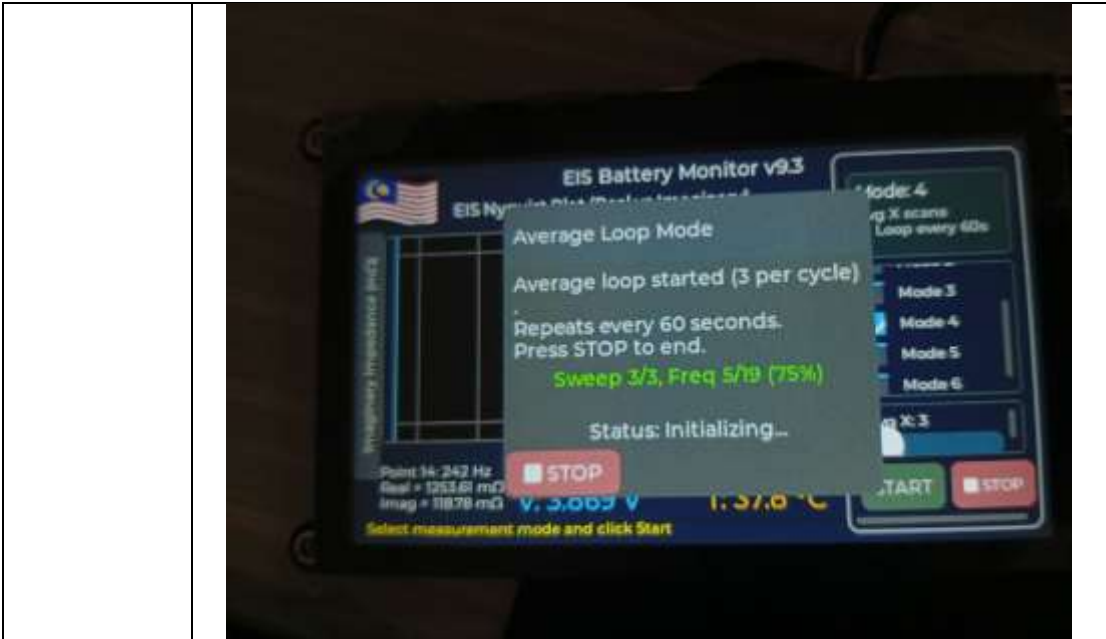


Figure 5.28: Average loop sweeps X of X

For each cycle, the system completed X full sweeps across the 19 frequencies (≈ 97 Hz to 4.943 kHz). At every frequency it collected one point per sweep and computed the average across the X repeats. After Sweep X/X reached Freq 19/19, the dialog switched to a 60-second countdown before the next cycle began automatically.

Waiting Interval

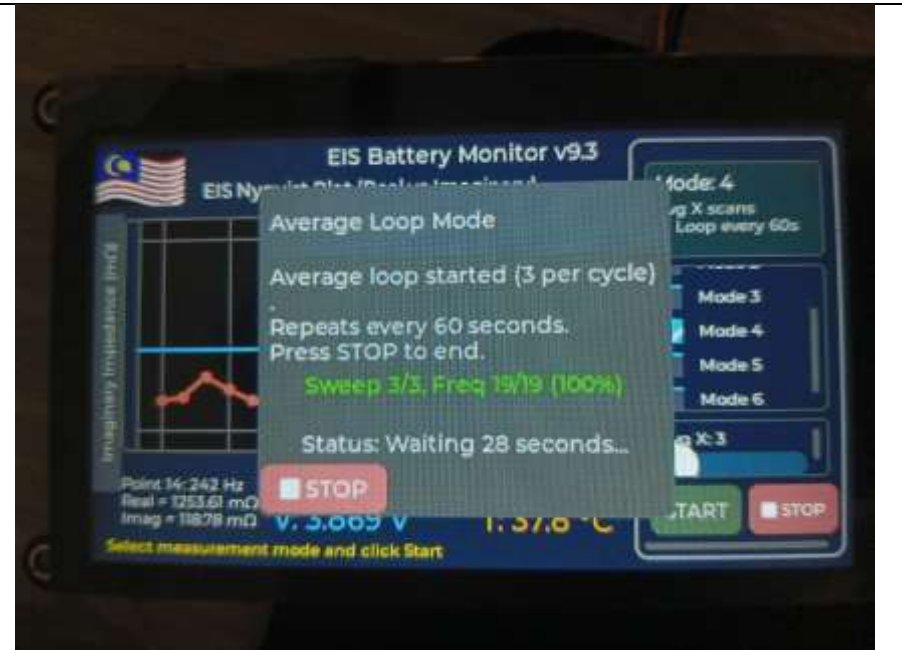

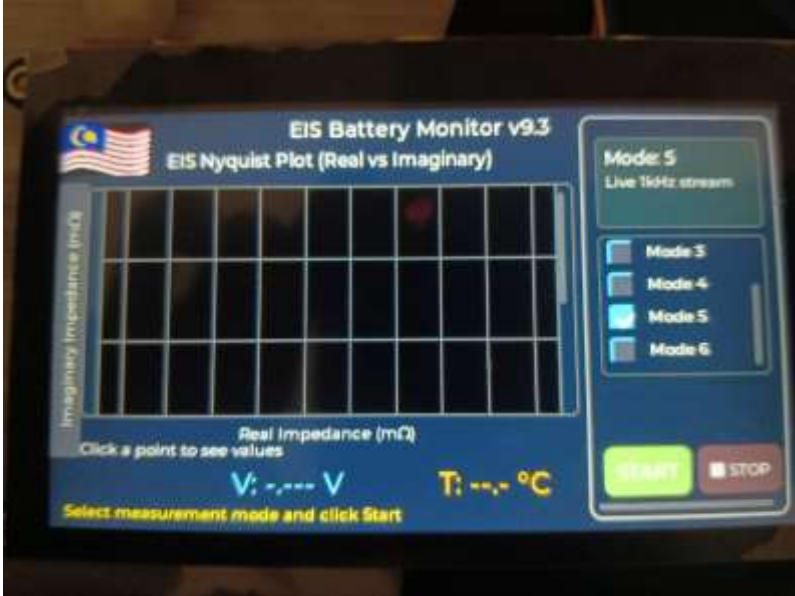
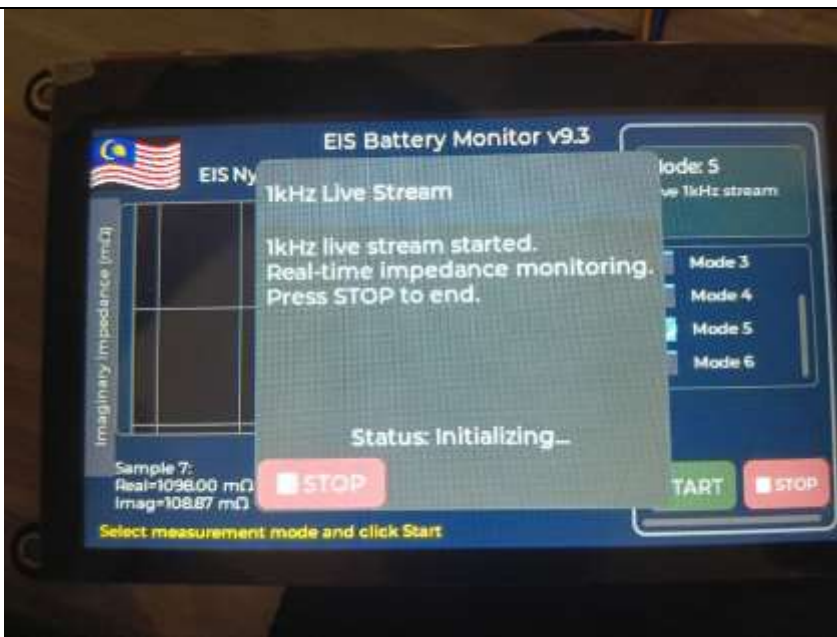

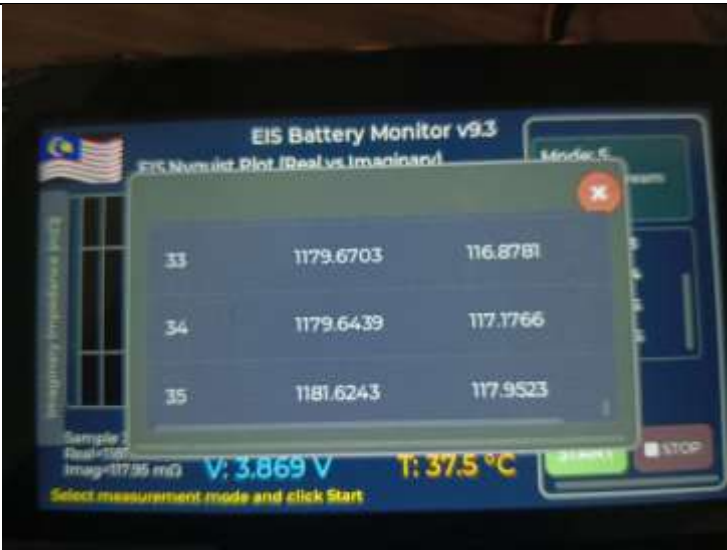


Figure 5.29: Countdown to next cycle

	When a cycle finished, the dialog reported the final sweep and frequency as complete and then showed a sixty-second countdown. At the next minute mark a new averaged cycle started automatically and the process continued until the user pressed Stop (<i>see Figure 5.d</i>).
Completion	 <p style="text-align: center;"><i>Figure 5.30: Averaged loop result</i></p> <p>During operation the plot reflected the most recent averaged curve and the on-screen readout allowed point inspection by tapping the trace. The readout displayed frequency together with real and imaginary impedance values in milliohms (<i>see Figure 5.e</i>).</p>
Outcome	Stopping the run closed the CSV safely and left the last curve on screen for review. The file was stored in /data/mode4 with a timestamped name so the user could adjust Avg X, choose another mode, or begin a new session without additional setup (<i>see Figure 5.f</i>).

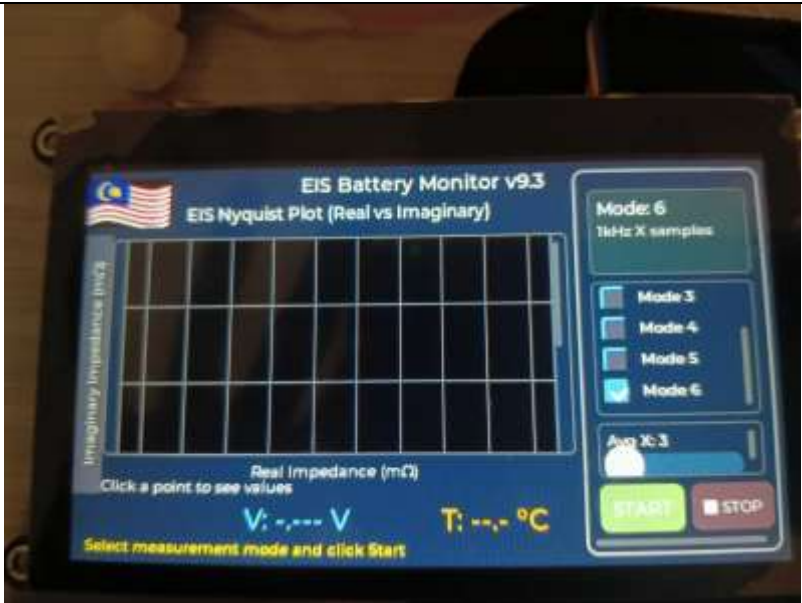
Mode 5

<p>Selection and setup</p>	 <p><i>Figure 5.31: Mode 5 selection screen</i></p> <p>The user selected Mode 5 on the right panel. The header confirmed “Mode 5 Live 1kHz stream”. (see Figure 5.xc)</p>
<p>Live Streaming</p>	 <p><i>Figure 5.32: Live stream started</i></p>

	<div></div> <p><i>Figure 5.33: Streaming samples in real time</i></p> <p>At a fixed 1 kHz rate the device received impedance samples and updated the display continuously. The dialog showed a running counter such as Sample 27 streaming and the plot advanced with each new point.</p>
Completion	Pressing Stop ended the stream immediately and the logger closed the file safely. The last data remained on screen so the user could review the trend.
Inspection	<div></div> <p><i>Figure 5.34: Live sample table view</i></p>

	The user could open the sample table to view the latest points numerically. Each row listed the sample index with Real and Imaginary impedance in ($m\Omega$) which matched the plotted trace for quick verification.
Outcome	The user could now select another mode, adjust the averaging count, or start a new session. The saved file was placed automatically under /data/mode5 with a timestamped name for later analysis.

Mode 6

Selection and setup	 <p><i>Figure 5.35: Mode 6 selection with Avg X</i></p> <p>The user selected Mode 6 on the right panel. The header confirmed “Mode 6 1kHz X samples”. (see <i>Figure 5.xc</i>)</p>
----------------------------	---

Averaging sequence



Figure 5.36: 1 kHz Average Dialog Initializing

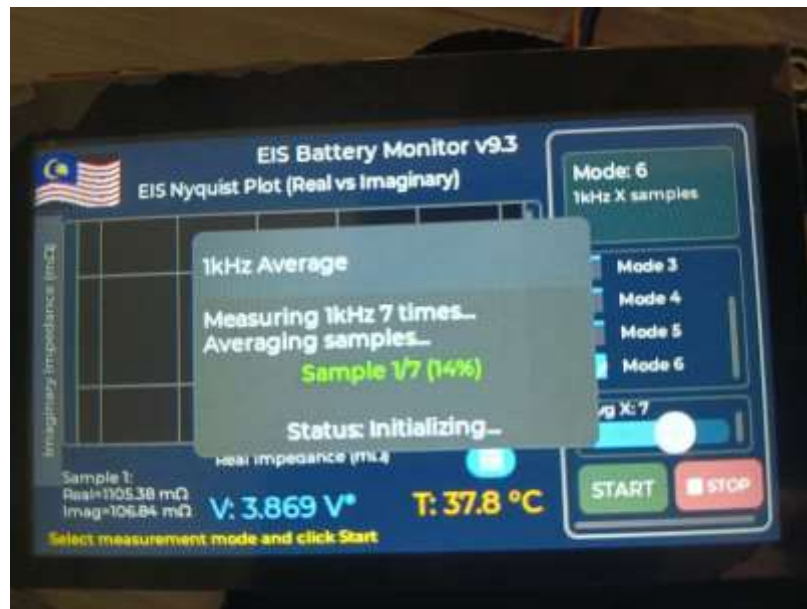




Figure 5.37: Sample 1 of X in progress

After Start the controller initialized the DNB110xA and began collecting 1 kHz samples. A modal panel reported the plan, for example “Measuring 1 kHz seven times, averaging samples,” and then advanced through the sequence with a counter such as Sample 1 of 7 and Sample 2 of 7 (see Figure 2 and Figure 3).

Completion	<div><p>The screenshot shows the 'EIS Battery Monitor v9.3' interface. A central dialog box displays '1kHz Average', 'Measuring 1kHz 7 times...', 'Averaging samples...', 'Sample 7/7 (100%)', and 'Status: Complete!'. The background shows a Nyquist plot grid with 'Imaginary Impedance (mΩ)' on the y-axis and 'Real Impedance (mΩ)' on the x-axis. At the bottom, it shows 'Sample 7: Real=1098.00 mΩ, Imag=108.87 mΩ', 'V: 3.869 V*', 'T: 37.8 °C', and a 'START' button.</p></div> <p><i>Figure 5.38: Sample X of X complete</i></p> <p>Sampling continued until the target count was reached. The dialog showed Sample X of X with a percentage and then changed the status to Complete to confirm that averaging had finished successfully (see Figure 4).</p>
Inspection	<div><p>The screenshot shows the same 'EIS Battery Monitor v9.3' interface. The Nyquist plot now displays a single green data point in the lower-left quadrant. The axes and bottom status information remain the same as in Figure 5.38.</p></div> <p><i>Figure 5.39: Averaged point plot in chart</i></p>

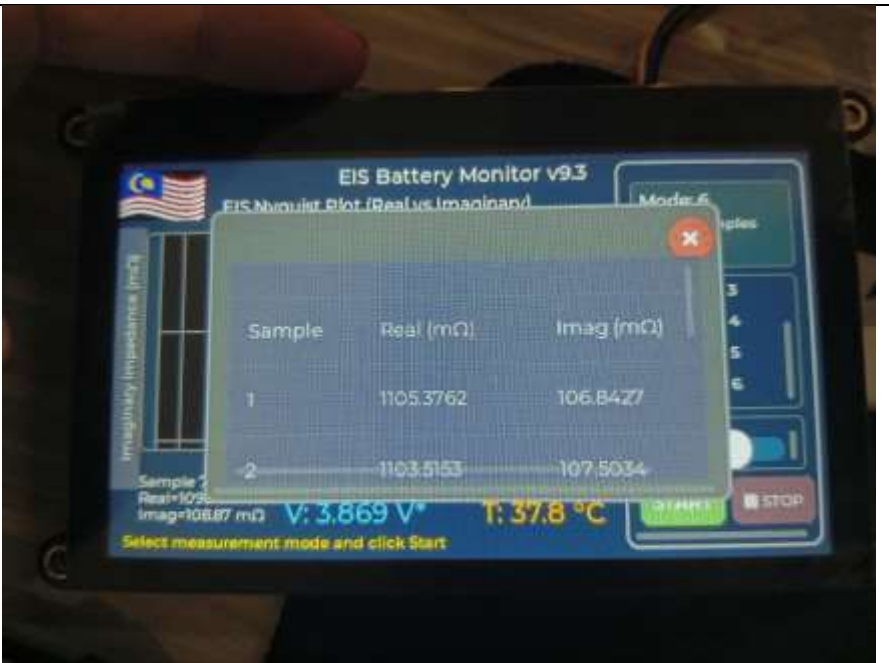


Figure 5.40: Averaging table page 1



Figure 5.41: Averaging table page 2

When sampling was in progress or after completion the user could open the table to review the individual measurements numerically. The table listed each sample together with Real and Imaginary impedance in milliohms so the user could verify stability across

	repeats (<i>see Figure 5, Figure 6, and Figure 7</i>). The averaged point was drawn on the canvas for visual confirmation.
Outcome	The user could now select another mode, adjust the averaging count, or start a new session. The saved file was placed automatically under /data/mode6 with a timestamped name for later analysis.

5.5 Implementation Issues and Challenges

A major challenge encountered was chart scaling and readability. Fixed axis ranges often made Nyquist plots appear either cramped or empty, depending on the cell under test. To address this, the display logic was enhanced to compute dynamic ranges with soft clamps and padded limits, while maintaining a fixed aspect ratio so arcs and circles remained visually accurate. Axes were clearly annotated in $\text{m}\Omega$, and a custom omega (Ω) glyph was introduced to improve clarity.

Another issue involved timing in Modes 3 and 4. The original implementation relied on blocking waits for the sixty-second dwell period, which froze animations and introduced clock drift when logging took longer than expected. This was resolved by adopting a millisecond tick with a countdown widget that updated smoothly while background tasks managed file writes. Users could also interrupt the countdown safely, with the system closing files properly to protect log integrity.

Memory constraints surfaced when charts, fonts, and images were combined, leading to linker errors and occasional runtime crashes. Stability was restored by switching to a Huge APP partition, enabling PSRAM, trimming non-essential assets, and resizing task stacks according to profiling results. These measures preserved double buffering and allowed the logger to sustain bursts without failure.

Responsiveness was also affected when plotting and SD card writes competed for CPU and SPI resources. The user interface was refactored to perform incremental redraws of only dirty regions, prebuilt labels were introduced to minimize string formatting, and a queued worker handled storage operations in batches. Touch inputs were filtered and debounced for accuracy, resulting in smooth dialogs, responsive point inspection, and uninterrupted interaction even during heavy activity.

Finally, coordination between the two ESP32-S3 cores was optimized. The user interface executed on one core with higher priority, while logging and peripheral drivers operated on the other. Shared resources such as the SD card and serial port were managed with mutexes to prevent conflicts. Stress testing confirmed reliable 1 kHz streaming, consistent sweep performance, and stable behavior across error conditions, with clear on-screen explanations for users. Overall, the primary challenges in scaling, timing, memory, responsiveness, and concurrency were systematically mitigated through dynamic resource management, non-blocking operations, careful buffer allocation, and a clear division of tasks between the two cores.

5.6 Concluding Remark

The implementation of the portable EIS measurement device successfully combined hardware integration and firmware development into a functional and reliable system. The ESP32-S3, DNB110xA module, JC4827W543 display, and microSD logging worked coherently to deliver real-time impedance measurements with live visualization and structured CSV output. Challenges in chart readability, timing, memory, responsiveness, and concurrency were systematically addressed through dynamic scaling, non-blocking timing models, optimized memory allocation, incremental rendering, and cross-core task separation. As a result, the system operates smoothly across all six modes, maintains stable logging, and provides a responsive and user-friendly interface. The completed implementation forms a strong foundation for accurate measurement, long-term stability, and future extension toward advanced data analysis and battery health prediction.

CHAPTER 6

System Evaluation and Discussion

In this chapter we present the evaluation of the portable EIS measurement device, including the testing methodology, performance metrics, and key results. We discuss the testing setup and findings, highlight challenges observed during validation, and assess how well the system met its objectives for accuracy, responsiveness, stability, and data logging. We conclude with a brief reflection on limitations and directions for future improvement.

6.1 System Testing and Performance Metrics

This section evaluated the portable EIS device using on-board checks, Serial Monitor traces, LVGL screen observations, and screenshots of the CSV files saved to the microSD card. The test scope covered the ESP32-S3 controller, the DNB110xA impedance front end, the JC4827W543 touch display with LVGL v9.3, and the automatic CSV logger. The aim was to verify functional correctness, responsiveness, logging integrity, and stability across all six operating modes.

The system testing includes:

- **Measurement sequence and sampling**

Each sweep captured nineteen frequencies from about 97 Hz to 4.943 kHz. Mode 5 streamed 1kHz samples and Mode 6 collected 1kHz samples for averaging. The firmware confirmed that frequency indices advanced in order and that every record contained valid real and imaginary impedance values in milliohms.

- **Repeatability**

Repeated runs on the same cell were overlaid on the Nyquist plot and compared by eye. Stability was also checked in the sample table to ensure similar ranges across sessions.

- **Throughput in Mode 5/6**

1kHz streaming was verified by counting samples per second from serial timestamps and by confirming that the CSV row count matched the run duration multiplied by one thousand within a small tolerance.

- **Timing correctness in Modes 3 and 4**

The dwell interval between cycles held close to sixty seconds by comparing the on-screen countdown with millisecond timestamps printed at the start and end of each cycle.

- **Logging integrity**

Every session produced a timestamped CSV in the correct mode folder under /data. Files opened correctly in a spreadsheet, contained only real_Impedance and imag_Impedance as designed, and closed cleanly on completion or when Stop was pressed. A screenshot of a CSV excerpt was captured as evidence.

6.2 Testing Setup and Result

Testing was carried out on the assembled device using the ESP32-S3 controller, the DNB110xA measurement front end, the JC4827W543 touch display with LVGL v9.3, and a FAT32 SD card. The device ran on USB power on a non-conductive surface. Firmware and UI settings followed the configuration described in Chapter 5. For each mode, serial monitor captured the real and imaginary impedance in $m\Omega$, the battery voltage, temperature, and auto-scaling Nyquist plots. Screenshots documented the on-screen dialogs, countdowns, plots, and tables. The SD card directory view and CSV excerpts were also captured as evidence of logging. All results below refer to these materials.

6.2.1 Timing correctness in Modes 3 and 4

```
Countdown: 56 seconds remaining
Countdown: 55 seconds remaining
Countdown: 54 seconds remaining
Countdown: 53 seconds remaining
Countdown: 52 seconds remaining
Countdown: 51 seconds remaining
Countdown: 50 seconds remaining
Waiting: 50 seconds remaining
```

Figure 6.1: Serial Monitor countdown confirming the one-minute dwell timer.



Figure 6.2: On-screen dialog showing the same waiting seconds for Mode 3/4.

The sixty-second dwell between cycles held steady under load. Start and end of each rest period were stamped to Serial and matched the on-screen countdown with an error within roughly one second over many repetitions. The interface stayed responsive during the countdown and immediately transitioned into the next sweep when the timer reached zero.

6.2.2 Logging integrity shown by CSV evidence

	A	B	C
1	RealImped	ImagImpedance_mOh	
2	281.2055	35.4375	
3	281.259	35.4317	
4	252.5179	24.2199	
5	257.1692	27.5622	
6	262.0144	30.4366	
7	266.0875	29.1552	
8	269.7748	36.9212	
9	274.4286	36.2016	
10	278.1171	35.6791	
11	259.0053	27.4587	
12	285.3022	35.125	
13	288.6056	35.2184	
14	292.296	35.1064	
15	295.195	35.1949	
16	298.3063	35.2722	
17	301.5921	35.4735	
18	304.1047	35.6789	
19	307.0236	35.8971	
20	309.5562	36.1959	

Figure 6.3: CSV excerpt: third-frequency record showing Real/Imag values used for cross-check.

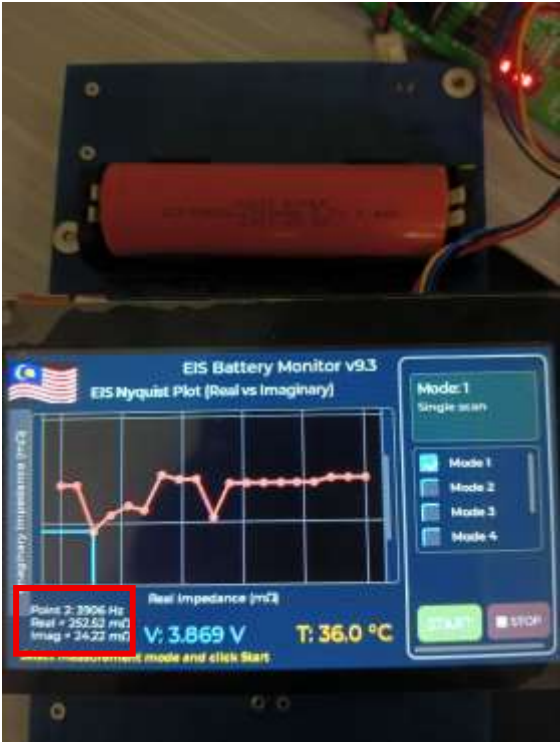


Figure 6.4: Device display: the third-frequency point reports the same Real/Imag values as Figure 6.3.

To demonstrate logging integrity a point-by-point cross-check was performed. From the Figure 6.3 and Figure 6.4 the third frequency shown on the device matched the third row in the CSV with the identical real and imaginary values.

6.2.3 Throughput and table-to-CSV cross-check in Modes 5 and 6

	A	B	C
1	# Mode 6: 1 kHz Individual Samples		
2	ReallImped ImagImpedance_mOhm		
3	1	278.6572	35.7407
4	2	278.6049	35.7426
5	3	278.5467	35.7537
6	4	278.4831	35.7659
7			



Figure 6.5: Table on device matches the corresponding CSV rows for the listed 1 kHz samples.

The verification focused on agreement between the on-screen table and the CSV log. The device collected at 1kHz, computed the average, and showed each sample’s real

and imaginary impedance in $m\Omega$. The corresponding CSV slice contained the same sample count in the same order, and

6.2.4 Repeatability

To examine repeatability and the ability to separate cells, two batteries were tested in repeated runs. Plots from the same battery overlaid closely and the point-inspection readout remained within a narrow range at representative frequencies, indicating stable behavior from run to run. Curves from the two batteries were visually distinct, with the blue cell and the orange cell forming different arcs or clusters on the Nyquist chart. This confirmed that the device could reproduce measurements on the same subject and could also differentiate subjects with different impedance characteristics.

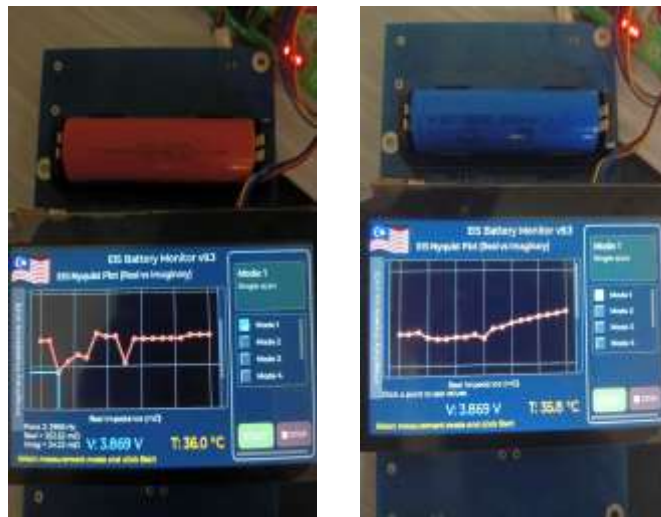


Figure 6.6: Nyquist plots for two cells

6.3 Project Challenges

The development of the portable EIS device required tight integration of the ESP32-S3 firmware, DNB110xA module, LVGL interface, and SD logging, with several challenges addressed during testing. Nyquist plots were improved through auto-scaling with padded limits and a fixed aspect ratio, while timing issues in looped modes were solved by replacing blocking delays with a non-blocking countdown. Performance bottlenecks caused by contention between display updates and SD writes were mitigated using a producer-consumer logging queue and incremental redraws. Memory stability was restored by enabling PSRAM, trimming assets, and profiling stack sizes,

while core-level resource conflicts were resolved through mutexes and clear ownership rules. Finally, user recovery was improved with fault dialogs for missing SD cards, devices, or modes, ensuring responsiveness, stability, and data integrity.

6.4 Objectives Evaluation

The project meets its main goals by delivering a compact portable EIS device that acquires real and imaginary impedance together with voltage and temperature in real time while plotting results live and logging to microSD in CSV for later analysis and machine learning. The LVGL interface is responsive and readable with auto-scaled axes progress and countdown views and clear error prompts and the firmware performs both the 19-point sweep and 1 kHz modes reliably with consistent dwell timing and stable operation over long sessions. Future work will focus on cross-checking absolute accuracy against a benchtop reference adding simple on-device feature extraction and expanding datasets to refine SoC and SoH predictions.

6.5 Concluding Remark

The evaluation shows that the prototype delivers stable EIS acquisition, responsive real-time visualization, and dependable CSV logging across all modes. Auto-scaled Nyquist plots, consistent timing, and a smooth LVGL interface support repeatable measurements on the same cell and clear separation between different cells, indicating practical diagnostic value. Next steps include benchmarking against a benchtop reference, refining or extending the frequency set for specific chemistries, and adding lightweight on-device feature extraction to speed analysis and strengthen future ML-based SoC and SoH estimation.

CHAPTER 7

Conclusion and Recommendations

7.1 Conclusion

This project set out to design and develop a compact and portable electrochemical impedance spectroscopy device that can deliver high-accuracy, real-time diagnostics for lithium-ion batteries, and the results demonstrate that this goal has been met at prototype level. The system integrates an ESP32-S3 controller with the DNB110xA impedance front end and a 4.3-inch capacitive touch display, unifying acquisition, visualization, and logging into an appliance-like workflow that is simple to operate. Across six operating modes, the firmware executes a 19-point frequency sweep and 1 kHz sampling paths with stable timing, while the LVGL interface renders auto-scaled Nyquist curves and responsive dialogs that make the measurement state clear at a glance. Data integrity is preserved through background CSV logging to microSD with safe open and close procedures, allowing subsequent offline analysis and machine-learning-assisted health estimation.

The engineering work emphasized robustness in the face of embedded constraints. Early issues with chart readability, dwell-time drift, memory pressure, and UI lag were addressed through a combination of auto-scaling, non-blocking timing, PSRAM utilization, partition changes, and a producer-consumer logging model. Resource sharing between cores and peripherals was stabilized with lightweight synchronization and clear ownership rules, which in turn protected files and kept interaction smooth even during sustained writes. Repeatability tests showed close overlays on the same cell and visual separability across different cells, indicating that the device can both reproduce a measurement and discriminate between batteries with distinct impedance characteristics.

At a system level, the device provides a coherent path from raw impedance to actionable insight. Users can start a session, observe live plots and status, and retrieve clean CSV logs without manual file handling. The data layout, timing discipline, and UI feedback are consistent across modes, reducing operator error and simplifying documentation.

While the prototype already supports on-device examination through point readouts and tabulated values, the design keeps a clear line of sight to analytics by capturing impedance components needed for feature extraction, model training, and downstream state-of-charge and state-of-health estimation.

The work inevitably has limitations that frame the next steps. Absolute accuracy still needs to be benchmarked against a benchtop instrument across chemistries and states of charge to quantify traceable error bounds. The fixed frequency set serves the current use case but may need extension or adaptive selection to capture features in cells with different time constants. On-device analytics are intentionally minimal at this stage and can be expanded to include basic feature computation and lightweight models for faster screening. Despite these boundaries, the project demonstrates a balanced combination of practicality and technical depth: a low-cost, portable platform that acquires reliable EIS data, presents it clearly, and stores it safely, ready to support predictive maintenance and battery lifecycle studies.

In summary, the device delivers a complete and reproducible pipeline from measurement to visualization and logging, validates the feasibility of portable EIS on embedded hardware, and establishes a strong foundation for data-driven diagnostics. With targeted validation, modest hardware refinements, and incremental analytics, it can evolve into a dependable field tool for research labs, service technicians, and educational settings alike.

7.2 Recommendation

The most immediate recommendation is to perform a structured accuracy campaign against a calibrated benchtop impedance analyzer using multiple cells, temperatures, and states of charge. This should include repeated measurements at controlled environmental conditions and a statistical analysis of bias and variance for real and imaginary components across the full frequency set. Establishing traceable error bars will not only quantify performance but also guide firmware adjustments such as averaging depth, dwell timing, and frequency coverage to reduce uncertainty where it matters most.

A second area of focus is selective enhancement of the measurement plan. While the present 19-point sweep captures useful behavior, certain chemistries and aging states benefit from denser sampling in specific bands. Introducing a configurable profile that allows finer steps near characteristic arcs or peaks can improve feature quality without materially increasing session time. Complementary improvements to excitation amplitude control and noise rejection can further stabilize results in low-signal regions. These adjustments should be paired with automated sanity checks in firmware so that outliers are flagged and, when appropriate, remeasured within the same session.

On the analytics side, the project will benefit from a disciplined data pipeline that converts CSV logs into standardized datasets with metadata for temperature, state of charge, cycle count, and cell type. Basic features such as magnitude, phase, real-axis intercept estimates, and semicircle parameters can be computed on device for quick screening and again offline for model training and validation. A lightweight, quantized model deployed on the ESP32-S3 can then provide immediate health hints to the user, while a richer model running off device can support deeper analysis. Versioning the CSV schema and adding a minimal session manifest will improve reproducibility and compatibility as the software evolves.

Usability and reliability can be strengthened through small but meaningful refinements. Adding a guided setup page, contextual tips for mode selection, and optional localization will reduce operator error and broaden accessibility. Improving enclosure design, cable strain relief, and EMI awareness will make the device more tolerant of

field conditions. Runtime diagnostics that check SD health, free space, and queue backlogs will preempt data loss and support preventive maintenance of the instrument itself. Optional secure export paths, such as a USB mass-storage view or a companion script for batch ingestion, will streamline the transition from measurement to analysis.

Finally, it is advisable to plan for sustained maintainability. Source modularity is already in place and should be preserved with unit tests for parsing, logging, and basic plotting routines, together with regression tests using captured sessions. Continuous logging under stress, thermal soak tests, and long-duration loop modes will reveal corner cases before deployment. As the dataset grows, consider releasing an anonymized benchmark subset and documentation to enable independent verification and model replication. Taken together, these recommendations will mature the prototype into a predictable, trustworthy tool that not only measures well but also integrates cleanly into laboratory workflows and data-driven battery diagnostics.

REFERENCES

- [1] “AI-Driven Battery State-of-Charge Estimation using Electrochemical Impedance Spectroscopy,” *IEEE Conference Publication | IEEE Xplore*, Aug. 29, 2023. <https://ieeexplore.ieee.org/document/10454954>
- [2] J. Wu *et al.*, “Design of a portable electrochemical impedance spectroscopy measurement system based on AD5941 for lithium-ion batteries,” *Journal of Energy Storage*, vol. 84, p. 110856, Feb. 2024, doi: 10.1016/j.est.2024.110856.
- [3] X. Ye, T. Jiang, Y. Ma, D. To, S. Wang, and J. Chen, “A portable, low-cost and high-throughput electrochemical impedance spectroscopy device for point-of-care biomarker detection,” *Biosensors and Bioelectronics X*, vol. 13, p. 100301, Dec. 2022, doi: 10.1016/j.biosx.2022.100301.
- [4] M. Koseoglou, E. Tsioumas, D. Papagiannis, N. Jabbour, and C. Mademlis, “A novel On-Board Electrochemical Impedance Spectroscopy System for Real-Time Battery Impedance Estimation,” *IEEE Transactions on Power Electronics*, vol. 36, no. 9, pp. 10776–10787, Mar. 2021, doi: 10.1109/tpel.2021.3063506.
- [5] M. Messing, T. Shoa, and S. Habibi, “Estimating battery state of health using electrochemical impedance spectroscopy and the relaxation effect,” *Journal of Energy Storage*, vol. 43, p. 103210, Sep. 2021, doi: 10.1016/j.est.2021.103210.
- [6] I. Babaeiyazdi, A. Rezaei-Zare, and S. Shokrzadeh, “State of charge prediction of EV Li-ion batteries using EIS: A machine learning approach,” *Energy*, vol. 223, p. 120116, Feb. 2021, doi: 10.1016/j.energy.2021.120116.
- [7] M. Grossi, C. Parolin, B. Vitali, and B. Riccò, “Electrical Impedance Spectroscopy (EIS) characterization of saline solutions with a low-cost portable measurement system,” *Engineering Science and Technology an International Journal*, vol. 22, no. 1, pp. 102–108, Sep. 2018, doi: 10.1016/j.jestch.2018.08.012.
- [8] S. Thapa, H. Agarwal, V. Ganesh, and A. K. Sahu, “Electrochemical impedance spectroscopy analysis and thermal mapping of different cross-

REFERENCES

- sectional cathode channels in open-cathode polymer electrolyte membrane fuel cell stack,” *Journal of Power Sources*, vol. 613, p. 234967, Jun. 2024, doi: 10.1016/j.jpowsour.2024.234967.
- [9] W. Li, J. Chen, K. Quade, D. Luder, J. Gong, and D. U. Sauer, “Battery degradation diagnosis with field data, impedance-based modeling and artificial intelligence,” *Energy Storage Materials*, vol. 53, pp. 391–403, Aug. 2022, doi: 10.1016/j.ensm.2022.08.021.
- [10] Jha, Biplov, and Lin Dong. *Lifetime Improvement with Predictive Maintenance of Power Electronics Based on Remaining Useful Life Prediction*. 12 Feb. 2024, <https://doi.org/10.1109/tpec60005.2024.10472254>. Accessed 13 Dec. 2024.
- [11] B. A. Braz, C. S. Moreira, V. B. Oliveira, and A. M. F. R. Pinto, “Electrochemical impedance spectroscopy as a diagnostic tool for passive direct methanol fuel cells,” *Energy Reports*, vol. 8, pp. 7964–7975, Jun. 2022, doi: 10.1016/j.egyr.2022.06.045.
- [12] Y. Li, "Electrochemical Impedance Spectroscopy Analysis of Lithium Ion Battery Based on Equivalent Circuit Model," *2020 2nd International Conference on Applied Machine Learning (ICAML)*, Changsha, China, 2020, pp. 282-285, doi: 10.1109/ICAML51583.2020.00065.
- [13] Y. Zhang, Q. Tang, Y. Zhang, J. Wang, U. Stimming, and A. A. Lee, “Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning,” *Nature Communications*, vol. 11, no. 1, Apr. 2020, doi: 10.1038/s41467-020-15235-7.
- [14] Y. Zhang, Q. Tang, Y. Zhang, J. Wang, U. Stimming, and A. A. Lee, “Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using
- [15] Tech Briefs, “Precise diagnosis of electric vehicle batteries using small currents,” *Tech Briefs*, Oct. 30, 2024. <https://www.techbriefs.com/component/content/article/51955-precise-diagnosis-of-electric-vehicle-batteries-using-small-currents>
- [16] E. Buchicchio, D. A. Alessio, F. Santoni, and P. Carbone, “Dataset on broadband Electrochemical Impedance Spectroscopy of Lithium-Ion

REFERENCES

Batteries for Different Values of the State of Charge,” *Mendeley Data*, Jun. 2022, doi: 10.17632/mbv3bx847g.3.

